Vorwort, 1. Auflage

Dieses Buch beschäftigt sich mit einem Teilbereich der Mustererkennung, nämlich der Klassifikation von Mustern. Darunter wird verstanden, dass ein relativ einfaches Muster – z. B. ein gedrucktes Schriftzeichen oder ein isoliert gesprochenes Wort – als Ganzes und unabhängig von anderen Mustern genau einer von mehreren möglichen Klassen zugeordnet wird. Jede Klasse entspricht dabei einer bestimmten Bedeutung. Zwar hat in den letzten Jahren die automatische Auswertung immer komplizierterer Muster, wie zum Beispiel kontinuierlich gesprochener Sprache, Grauwertbilder und Bildfolgen, ein rasch zunehmendes Interesse gefunden, jedoch gehört die Klassifikation nach wie vor zu den grundlegenden Techniken. Diese Tatsache wird auch dadurch unterstrichen, dass es seit mehreren Jahren eine Reihe kommerzieller Geräte gibt, die auf Methoden aus dem Bereich der Klassifikation von Mustern basieren.

Für die Lektüre des Buches werden Grundkenntnisse der Höheren Mathematik und Wahrscheinlichkeitsrechnung vorausgesetzt. Es wendet sich an Wissenschaftler, die diese Techniken als Hilfsmittel bei ihrer Arbeit einsetzen möchten, und an Studenten, die sich intensiver mit diesem Problem der automatischen Informationsverarbeitung beschäftigen möchten. Der Inhalt des Buches kann etwa im Rahmen einer einsemestrigen Vorlesung behandelt werden. Das einführende erste Kapitel enthält die wichtigsten Begriffe und gibt eine Abgrenzung des behandelten Stoffs. Im nächsten Kapitel wird auf die wichtigsten Verfahren der Vorverarbeitung von Mustern eingegangen. Im Prinzip geht es dabei um eine Vereinfachung der nachfolgenden Verarbeitung. Das zentrale Problem bei der Klassifikation, nämlich die Ermittlung von Merkmalen, welche die für die Klassenzugehörigkeit wesentliche Information enthalten, wird im dritten Kapitel behandelt. Die Kapitel vier und fünf beschäftigen sich mit der eigentlichen Klassifikation der extrahierten Merkmale; wegen der beiden Möglichkeiten, als Werte von Merkmalen entweder reelle Zahlen oder Symbole zu verwenden, erfolgt hier eine Aufspaltung in zwei Kapitel. Der gesamte Inhalt ist nach Verarbeitungsmethoden gegliedert und nicht nach speziellen Anwendungen oder Problemen. Nur im sechsten Kapitel wird ganz kurz skizziert, wie bestimmte Methoden zur Lösung einer konkreten Aufgabe eingesetzt wurden.

Dem Springer Verlag, vertreten durch Herrn G. Rossbach, sei an dieser Stelle für die Herausgabe des Buches und die Unterstützung bei der Reinschrift eines Teils des Manuskripts gedankt. Der größte Teil des Manuskripts wurde von Frau S. Zett geschrieben, die Zeichnungen von Herrn A. Cieslik angefertigt; beiden danke ich für ihre sorgfältige Arbeit.

H. Niemann
Erlangen, Mai 1983

Vorwort, 2. Auflage

Abgesehen davon, dass in dieser neuen Auflage des Buches von 1983 die Gestaltung durch Verwendung von LaTeX verbessert wurde, wurden auch einige inhaltliche Ergänzungen und Überarbeitungen vorgenommen. Weiterhin wurde die Veröffentlichung im Internet gewählt, um rascher Korrekturen oder Ergänzungen vornehmen zu können; für die Genehmigung dazu wird dem Springer Verlag, vertreten durch Frau Hanich, gedankt.

Neu aufgenommen wurde im Kapitel *Einführung* eine kurze Charakterisierung typischer Klassifikationsaufgaben sowie eine kurze Darstellung wichtiger Optimierungsverfahren. Im Kapitel *Vorverarbeitung* wurden Abschnitte über Vektorquantisierung, Beispiele für lineare Filter, morphologische Operationen, Diffusionsfilter und Interpolation ergänzt. Das Kapitel „Merkmale“ wurde um Abschnitte über Wavelets, Filterbänke, Merkmale für Textur–, Sprach– und Objekterkennung sowie Invarianten erweitert. Gestrichen wurde hier der Abschnitt über Beispiele für Merkmale. Im Kapitel *Numerische Klassifikation* wurde die Einführung des optimalen Klassifikators neu gefasst und Abschnitte über neuronale Netze, Support Vektor Maschinen, Sprach– und Objekterkennung eingefügt. Gestrichen wurde hier der Abschnitt über Toleranzgebiete für die Klassifikation. Das Kapitel über *Nichtnumerische (Syntaktische) Klassifikation* wurde gestrichen, weil sich hier in den letzten Jahren wenig Neues ergeben hat und weil für Klassifikationsaufgaben sich mehr und mehr die numerischen Verfahren durchgesetzt haben. Die Literaturangaben wurden generell im letzten Abschnitt jedes Kapitels zusammengefasst.

Je Kapitel (TEX \chapter) wurde eine Versionsnummer VK.i.j.k, je Abschnitt (TEX \section) eine Versionsnummer VA.i.j.k eingeführt. In VA wird der Index i erhöht, wenn inhaltliche Ergänzungen im Umfang von mindestens einem Unterabschnitt (TEX \subsection) vorgenommen wurden; der Index j wird erhöht, wenn sachliche Fehler korrigiert oder kleinere inhaltliche Modifikationen oder Ergänzungen gemacht wurden; der Index k wird erhöht, wenn Druckfehler korrigiert wurden. Änderungen in VA ziehen analoge Änderungen in VK nach sich.

Mein besonderer Dank gilt den Herren Fentze und Popp sowie den Studentinnen und Studenten, die mich bei der Erstellung der LaTeX-Version und der Anfertigung von Bildern unterstützt haben. Weiterhin danke ich den Mitarbeiterinnen und Mitarbeitern, die für mich immer wieder geeignete “styles” gesucht und gefunden haben.

H. Niemann
Erlangen, Mai 2003

Dank

Der Autor dankt für Hinweise auf Druckfehler und für Verbesserungsvorschläge:

J. Adelhardt
R. Chrastek
C. Derichs
B.S. Deutsch
R. Deventer
K. Donath
C. Gräßl
M. Grzegorzek
C. Hacker
T. Haderlein
M. Levit
B. Ludwig
E. Nöth
J. Schmidt
I. Scholz
S. Steidl
G. Stemmer
F. Vogt
M. Wacker
S. Wenhardt
U. Zeißler
T. Zinßer

Inhaltsverzeichnis

Kapitel 1

Einführung (VK.1.3.3, 16.03.2003)

Ein ernsthafter Mann, der sich mit ernsthaften Dingen beschäftigt, sollte nicht schreiben. (PLATON)

Man weiß oder ahnt: wenn das Denken nicht rein und wach und die Verehrung des Geistes nicht mehr gültig ist, dann gehen bald auch die Schiffe und Automobile nicht mehr richtig, dann . . . kommt das Chaos. (HESSE)

In der Einführung wird zunächst der Begriff „Mustererkennung“ im weiten Sinne definiert und die allgemein zugrunde liegenden Prinzipien erläutert. Dann folgt eine Konzentrierung auf das Teilgebiet der Klassifikation von Mustern, das im vorliegenden Buch ausschließlich behandelt wird. Die grundsätzliche Vorgehensweise wird skizziert und typische Klassifikationsprobleme werden vorgestellt. Über Optimierungsverfahren als wesentliches Element systematischer Lösungsansätze wird ein kurzer Überblick gegeben. Einige Anwendungsgebiete werden vorgestellt.

1.1 Allgemeines

Worte können die Wahrheit nie so darstellen, wie sie ist. (TAHUI)

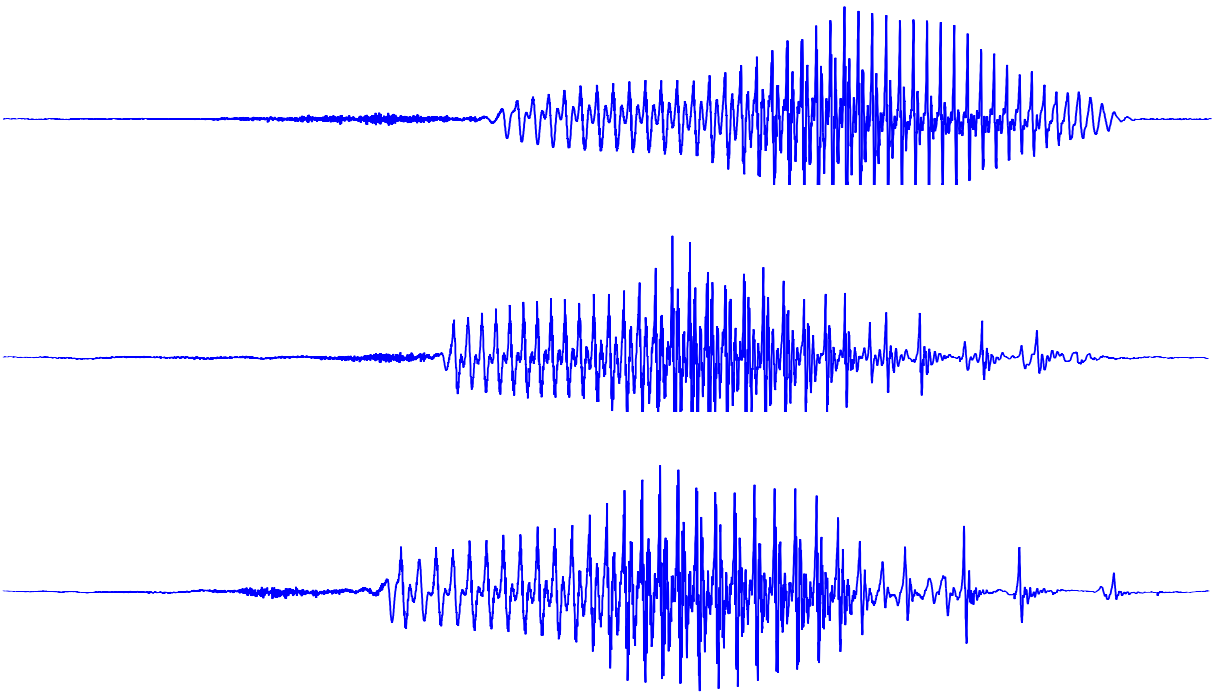
Worte vermitteln die Wahrheit, wenn sie richtig verstanden werden. (MIAO–HSI)

Mit der Entwicklung von Digitalrechnern, deren Leistungsfähigkeit in den letzten Jahrzehnten ständig erhöht wurde und auch in den nächsten Jahren weiter gesteigert werden wird, ist die Möglichkeit gegeben, äußerst komplizierte Prozesse der Informationsverarbeitung zu untersuchen, zu modellieren und zu simulieren. Eine interessante und wichtige Form der Informationsverarbeitung sind die perzeptiven Fähigkeiten von Lebewesen, insbesondere von Wirbeltieren. Zur *Perzeption* wird hier das Bemerken, Auswerten und Interpretieren von Sinneseindrücken gerechnet, wobei für den Menschen optische und akustische Eindrücke besonders wichtig sind. Jede zielgerichtete menschliche Aktivität erfordert Perzeption, und jeder ist in der Lage, ungeheure Mengen von Sinneseindrücken zu verarbeiten. Trotzdem läuft diese Verarbeitung weitgehend unbewusst ab, und die dabei erforderlichen Operationen und Algorithmen sind weitgehend unbekannt. Das wird spätestens dann deutlich, wenn man versucht, einige perzeptive Leistungen beispielsweise durch ein Rechnerprogramm zu simulieren. Die schwierigen Probleme bei der Erforschung menschlicher (und maschineller) Perzeption werden aus folgendem Zitat deutlich:

> Recently I was trying to explain to an intelligent woman the problem of understanding how it is that we perceive anything at all, and I was not having any success. She could not see why there was a problem.
>
> Finally in despair I asked her how she herself thought she saw the world. She replied that she probably had somewhere in her head something like a little television set. "So who", I asked "is looking at it?" — She now saw the problem immediately.
>
> F.H.C. Crick: Thinking About the Brain. Scientific American 241, No. 3. (1979) 181-188.

Die Untersuchung der mathematisch–technischen Aspekte der Perzeption ist nicht nur von wissenschaftlichem Interesse, vielmehr verspricht ein gründlicheres Verständnis derselben zahlreiche Anwendungsmöglichkeiten, von denen einige im Abschnitt 1.7 genannt werden.

Forschungs– und Entwicklungsaktivitäten, welche die mathematisch–technischen Aspekte der Perzeption betreffen, sind das Gebiet der **Mustererkennung** im weiten Sinne. Einige damit zusammenhängende Begriffe werden im nächsten Abschnitt genauer definiert. Dagegen werden mathematisch–biologische Aspekte hier nicht betrachtet, da sie in den Bereich der Biokybernetik, Physiologie und Psychologie gehören. Die Frage, ob Maschinen überhaupt zur Perzeption fähig sind, ist hier belanglos. Es steht außer Frage, dass Perzeption möglich ist, wie von den Organismen demonstriert wird. Bisher ist kein Naturgesetz bekannt, welches die Simulation von perzeptiven Leistungen durch Maschinen ausschließt. Es sei betont, dass es in der Mustererkennung vorrangig um die *Simulation* einer perzeptiven Leistung geht und weniger um die Modellierung oder Kopierung der dafür in Organismen eingesetzten Algorithmen. Beispielsweise kommt es also darauf an, gesprochene Sprache mit einer Maschine *ähnlich zuverlässig* zu erkennen wie ein Mensch; aber es kommt nicht darauf an, es *genauso* wie der Mensch zu machen, also Ohr und Sprachzentrum des Gehirns mit Maschinen möglichst genau zu modellieren. Ein Standardbeispiel in diesem Zusammenhang sind Vögel und Flugzeuge: Erstere demonstrieren, dass Fliegen möglich ist, beide nutzen das physikalische Prinzip des Auftriebs, aber das Antriebsverfahren ist bei beiden völlig verschieden („Flugzeuge schlagen nicht mit den Flügeln").



1.2 Definitionen

Die Welt ist alles, was der Fall ist. Was der Fall ist, die Tatsache, ist das Bestehen von Sachverhalten. Das logische Bild der Tatsachen ist der Gedanke. (WITTGENSTEIN)

Auch wenn nur *eine* vereinheitlichte Theorie möglich ist, so wäre sie doch nur ein System von Regeln und Gleichungen. Wer bläst den Gleichungen den Odem ein und erschafft ihnen ein Universum, das sie beschreiben können? (HAWKING)

Nachdem im vorigen Abschnitt eine allgemeine Darstellung des Ziels der Mustererkennung gegeben wurde, werden nun einige wichtige Begriffe genauer definiert und durch Beispiele erläutert. Gegenstand der Perzeption sind Eindrücke aus der Umwelt.

Definition 1.1 *Für die Zwecke der Perzeption genügt es, die* **Umwelt** *als die Gesamtheit der physikalisch* messbaren *Größen aufzufassen, die formal durch die Menge*

$$U = \{{}^{\varrho}\boldsymbol{b}(\boldsymbol{x}) | \varrho = 1, 2, \ldots\} \tag{1.2.1}$$

der messbaren Größen oder Funktionen ${}^{\varrho}\boldsymbol{b}(\boldsymbol{x})$ *dargestellt wird.*

Offensichtlich lässt sich jedes Objekt und jedes Ereignis durch genügend viele geeignet gewählte Funktionen beschreiben. Der Funktionswert gibt für jeden Punkt des Raumes und/oder der Zeit eine charakteristische Größe an. Beispielsweise können einige Eigenschaften eines festen Körpers durch Angabe seiner Dichte in jedem Punkt des Raumes charakterisiert werden, und falls erforderlich kann man diese Angaben durch Daten über die chemische Zusammensetzung, das Lichtreflektionsvermögen der Oberfläche und andere mehr erweitern. Ein anderes Beispiel ist die Angabe des zeitlich und örtlich veränderlichen elektrischen Feldstärkevektors einer elektromagnetischen Welle. Da U alle Funktionen enthalten soll, muss die Zahl der Komponenten von $\boldsymbol{b}$ und $\boldsymbol{x}$ offen bleiben. Sie kann für jeden Wert von ϱ unterschiedlich sein. Es gibt kein biologisches oder technisches System, das die ganze Umwelt erfassen kann. Sinnesorgane

und Messinstrumente reagieren stets nur auf Ausschnitte. Zum Beispiel erfasst das menschliche Auge trotz seiner enormen Leistungsfähigkeit nur einen kleinen Teil aus dem Spektrum der elektromagnetischen Wellen. Ein universelles technisches System für die Mustererkennung, das die ganze Umwelt oder auch nur einen großen Teil davon aufnehmen und verarbeiten könnte, ist zur Zeit nicht denkbar und ist in jedem Falle unwirtschaftlich und uneffektiv. Daher ist es zweckmäßig, sich auf einen bestimmten Problemkreis zu beschränken.

Definition 1.2 *Ein* **Problemkreis** Ω *enthält nur Objekte (Funktionen) eines bestimmten und begrenzten Anwendungsgebietes, die mit geeigneten Sensoren erfasst werden.*

Wenn Muster nur mit einem *Sensor aufgenommen werden, ist der Problemkreis gegeben durch eine Menge*

$$\Omega = \{{}^{\varrho}\boldsymbol{f}(\boldsymbol{x}) | \varrho = 1, 2, \ldots\} \subset U \tag{1.2.2}$$

von Funktionen ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x})$ *und ist eine Untermenge der Umwelt U.*

Wenn mehrere *Sensoren* $1, \ldots, \sigma, \ldots, s$ *eingesetzt werden, ist der Problemkreis gegeben durch eine Menge*

$$\Omega = \{({}^{\varrho}\boldsymbol{f}_1(\boldsymbol{x}_1), \ldots, {}^{\varrho}\boldsymbol{f}_\sigma(\boldsymbol{x}_\sigma), \ldots, {}^{\varrho}\boldsymbol{f}_s(\boldsymbol{x}_s)) \,|\varrho = 1, 2, \ldots\} \subset U \ . \tag{1.2.3}$$

Wenn auch ein Vektor von individuellen *Parametern* ${}^{\varrho}\boldsymbol{\Xi}_\sigma$ *der Aufnahmebedingungen notwendig ist (z. B. beim aktiven Sehen), definieren wir den Problemkreis zu*

$$\Omega = \{([{}^{\varrho}\boldsymbol{f}_1(\boldsymbol{x}_1), {}^{\varrho}\boldsymbol{\Xi}_1], \ldots, [{}^{\varrho}\boldsymbol{f}_s(\boldsymbol{x}_s), {}^{\varrho}\boldsymbol{\Xi}_s]) \,|\varrho = 1, 2, \ldots\} \subset U \ . \tag{1.2.4}$$

Im Unterschied zu (1.2.1) ist in (1.2.2) die Zahl der Komponenten die gleiche für alle ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x}) \in \Omega$, aber natürlich wird diese Zahl i. Allg. verschieden sein für verschiedene Problemkreise. Beispiele für Problemkreise sind die Klassifikation handgedruckter alphanumerischer Zeichen, die Prüfung der Echtheit (Verifikation) von Unterschriften, die automatische Ermittlung der Schaltelemente und Verbindungen in einem elektrischen Schaltplan, die Detektion und/oder Identifikation und Lokalisation von Gesichtern oder das Verstehen von gesprochenen Sätzen in deutscher Sprache. Jeder Problemkreis Ω erfordert entsprechende Geräte bzw. *Sensoren* zur Messung der darin vorkommenden Funktionen, und umgekehrt wird durch die Wahl der Sensoren eine Menge messbarer Größen, die ein Ausschnitt aus der Umwelt sind, bestimmt. Der Wahl der Sensoren kommt also eine ganz wesentliche Bedeutung für die weitere Verarbeitung zu. Die Verwendung mehrerer Sensoren, wie in (1.2.3) eingeführt, liefert Information über unterschiedliche Aspekte und ist daher in komplexen Problemen u. U. sinnvoll oder notwendig. Beispiele sind die Aufnahme des Sprachsignals und des Bildes der Lippenbewegungen für die Worterkennung oder von Röntgen– und Magnetresonanzbildern des Kopfes in der medizinischen Bildverarbeitung. Die Zahl der Komponenten der von verschiedenen Sensoren aufgenommenen Muster wird i. Allg. verschieden sein. Der letzte Fall (1.2.4) mit praktisch beliebigen Parametern für die Zahl der Komponenten von $\boldsymbol{f}$ und $\boldsymbol{x}$ sowie für die Aufnahmebedingungen je Muster wird der Vollständigkeit halber angeführt. Er wird im Rahmen dieses Buches nicht weiter betrachtet. Für viele praktisch interessante Klassifikationsprobleme ist (1.2.2) bereits hinreichend. Damit lässt sich nun definieren, was unter einem Muster zu verstehen ist, wobei wir uns auf den einfachen Fall des Problemkreises gemäß (1.2.2) beschränken.

Definition 1.3 *Die Elemente der Menge* Ω*, also die zu einem Problemkreis gehörigen Funktionen, heißen* **Muster**.

1) Ein Muster ist eine Funktion

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ f_2(x_1, \ldots, x_n) \\ \vdots \\ f_m(x_1, \ldots, x_n) \end{pmatrix} . \tag{1.2.5}$$

2) Äquivalent dazu lässt sich ein Muster auffassen als die Menge der Tupel

$$\mathrm{f} = \left\{ (\boldsymbol{x},\ \boldsymbol{f})^{\mathsf{T}} | \forall \boldsymbol{x},\ \forall \boldsymbol{f} \right\} = \left\{ (\boldsymbol{x},\ \boldsymbol{f}_{\boldsymbol{x}})^{\mathsf{T}} \right\} . \tag{1.2.6}$$

Die Frage, ob die Terminologie, die von „Muster“, „Mustererkennung“ und dergleichen mehr spricht, glücklich gewählt ist, sei hier zwar aufgeworfen, aber ihre Beantwortung, die eine vorherige lange und vermutlich langweilende Diskussion unterschiedlicher Definitionen erfordern würde, dem Leser anheimgestellt. Es ist aber zu erwähnen, dass Bezeichnungen wie Muster und Mustererkennung (englisch “pattern” und “pattern recognition”) international eingeführt und in der einschlägigen Fachliteratur üblich sind. Es ist auch zu erwähnen, dass es leider immer noch keine Definition des Begriffs „Muster“ gibt, die ähnlich präzise und mathematisch verwertbar ist wie die Definition der Information durch SHANNON. Ist das ein Hinweis darauf, dass es keine gibt?

Für einen bestimmten Problemkreis ist, wie erwähnt, die Zahl der Komponenten von $\boldsymbol{f}$ und $\boldsymbol{x}$ konstant, d. h. die Indizes m und n sind für alle ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x}) \in \Omega$ unveränderlich. Zum Beispiel besteht ein Vektorkardiogramm i. Allg. aus drei Zeitfunktionen $f_i(t)$, es ist also $m = 3$ und $n = 1$. Ein Farbfernsehbild besteht aus zeitveränderlichen Bildern $f_{\mathrm{r}}(x, y, t)$, $f_{\mathrm{g}}(x, y, t)$, $f_{\mathrm{b}}(x, y, t)$ in den drei Spektralbereichen rot, grün und blau, wobei es hier weniger wichtig ist, dass für die Fernsehübertragung i. Allg. noch eine andere Kodierung vorgenommen wird; es ist hier also $m = 3$, $n = 3$. Sprache und Geräusche, die von einem Mikrofon in einen elektrischen Spannungsverlauf umgewandelt wurden, bestehen nur aus einer Zeitfunktion $f(t)$ mit $m = n = 1$. Solche Muster werden auch als wellenförmige Muster bezeichnet. Ein übliches Schwarzweiß-Foto läßt sich als Funktion $f(x, y)$ darstellen, wobei der Funktionswert den Grauwert des Bildes an der Stelle (x, y) angibt; hier ist also $m = 1$ und $n = 2$. Diese Beispiele verdeutlichen, dass es kein Problem bereitet, die üblichen auditiven und visuellen Umwelteindrücke durch geeignete Funktionen darzustellen. Mit entsprechenden Aufnahmegeräten, sogenannten Multispektralabtastern, ist es auch möglich, Bilder in solchen Spektralbereichen aufzunehmen, in denen das Auge nicht empfindlich ist, z. B. im Infrarotbereich. Dabei ergeben sich deutlich mehr Komponenten von $\boldsymbol{f}$ als der rote, grüne und blaue Spektralkanal. Die hyperspektralen Bilder enthalten Aufnahmen in vielen, dicht beieinander liegenden, schmalen Wellenlängenbereichen, sodass die Wellenlänge λ hier als weitere unabhängige Variable in dem Muster $f(x, y, t, \lambda)$ eingeführt werden kann. Mit Computer– oder MR–Tomographen können „zerstörungsfrei“ Schnittbilder von Objekten hergestellt werden, deren Grauwerte von den Eigenschaften des geschnittenen Volumens abhängen. Es wird nun nochmals auf den Begriff Mustererkennung eingegangen.

Definition 1.4 *Die* **Mustererkennung** *beschäftigt sich mit den mathematisch–technischen Aspekten der automatischen Verarbeitung und Auswertung von Mustern. Es wird für ein physikalisches* Signal *(z. B. Sprache, Bild, Meßwert) eine geeignete* Symbolkette *(bzw. in eine formale*

Datenstruktur) berechnet. Dazu gehört sowohl die Klassifikation einfacher Muster *als auch die Klassifikation und Analyse* komplexer Muster*:*

$$\boldsymbol{f} \;\Rightarrow\; \begin{cases} \Omega_\kappa & : \text{ eine } \textit{Klasse} \text{ ,} \\ \boldsymbol{\Omega} = [{}^1\Omega, \ldots, {}^N\Omega] & : \textit{ eine} \text{ Folge } \textit{von Klassen} \text{ ,} \\ (\Omega_\kappa, \boldsymbol{t}_\kappa, \boldsymbol{R}_\kappa) & : \textit{ Klasse} \text{ und } \textit{Lokalisation} \text{ ,} \\ \mathcal{B} & : \textit{ symbolische Beschreibung} \text{ .} \end{cases} \tag{1.2.7}$$

Eine Veranschaulichung für die Beispiele der Sprach- und der Objekterkennung gibt Bild 1.2.1. Da der Begriff Muster sehr umfassend definiert wurde und auch die Begriffe Verarbeitung und Auswertung nicht weiter festgelegt wurden, ist damit Mustererkennung in einem weiten Sinne definiert. Eine Präzisierung erfolgt durch die Einführung der Teilbereiche Klassifikation und Analyse, die unten noch genauer erläutert werden. Mit der Unterscheidung zwischen *einfachen* und *komplexen* Mustern soll hier lediglich an die intuitiv einleuchtende Tatsache angeknüpft werden, dass beispielsweise ein einzelnes gedrucktes Schriftzeichen ein wesentlich einfacheres Muster ist als ein Farbfoto, oder ein isoliert gesprochenes Wort ein wesentlich einfacheres Muster als ein zusammenhängend gesprochener Satz. Dagegen ist nicht an eine quantitative Charakterisierung und die Festlegung einer scharf definierten Schwelle zwischen beiden gedacht. Ebenso soll die oben eingeführte Trennung zwischen Klassifikation und Analyse nicht implizieren, dass beide Operationen nichts miteinander zu tun haben; die Gemeinsamkeiten werden noch verdeutlicht werden. Zur weiteren Klärung wird zunächst der Begriff Klasse bzw. Musterklasse genauer betrachtet, danach der Begriff Klassifikation.

Definition 1.5 **Klassen** *oder* **Musterklassen** Ω_κ *ergeben sich durch eine* Zerlegung *der Menge* Ω *in* k *oder* $k+1$ *Untermengen* Ω_κ*,* $\kappa = 1, \ldots, k$ *oder* $\kappa = 0, 1, \ldots, k$*, sodass gilt*

$$\begin{aligned} \Omega_\kappa &\neq \emptyset \qquad \kappa = 1, \ldots, k \text{ ,} \\ \Omega_\kappa \cap \Omega_\lambda &= \emptyset \qquad \lambda \neq \kappa \text{ ,} \\ \textit{entweder} \quad \bigcup_{\kappa=1}^{k} \Omega_\kappa &= \Omega \qquad \textit{oder} \quad \bigcup_{\kappa=0}^{k} \Omega_\kappa = \Omega \text{ .} \end{aligned} \tag{1.2.8}$$

Für die Menge Ω gibt es viele Zerlegungen, die den obigen Anforderungen genügen, jedoch werden für den Anwender nur wenige, vielfach sogar *nur eine* praktisch interessant sein. Eine solche praktisch interessante Zerlegung ist dadurch gekennzeichnet, dass die Muster einer Klasse einander ähnlich und/oder die Muster verschiedener Klassen einander unähnlich sind. Eine geeignete, d. h. den Intentionen des Anwenders gerecht werdende, Definition der Ähnlichkeit wird dabei vorausgesetzt.

Eine Klasse enthält eine Teilmenge der Muster eines Problemkreises. Wenn z. B. im Zusammenhang mit der Klassifikation isoliert gesprochener Wörter die Klasse Ω_κ die Bedeutung „Haus“ hat, so gehören zu Ω_κ alle Muster – in diesem Falle alle Zeitfunktionen $f(t)$ – die entstehen, wenn verschiedene Sprecher zu verschiedenen Zeiten mit unterschiedlicher Lautstärke, Tonhöhe, Geschwindigkeit usw. das Wort Haus sprechen. In (1.2.8) wird gefordert, dass Klassen disjunkt sind. Das ist für viele Anwendungen angemessen, da z. B. eine Ziffer nicht gleichzeitig eine 7 und eine 1 sein kann. Wenn beide Interpretationen der Ziffer möglich sind, so sollte man sie zurückweisen, wofür oben die **Rückweisungsklasse** Ω_0 eingeführt wurde. Diese kann auch als eine Klasse angesehen werden, in die Muster, die andere statistische Eigenschaften als die zum Training des Klassifikators verwendeten haben, eingeordnet werden – die sog.

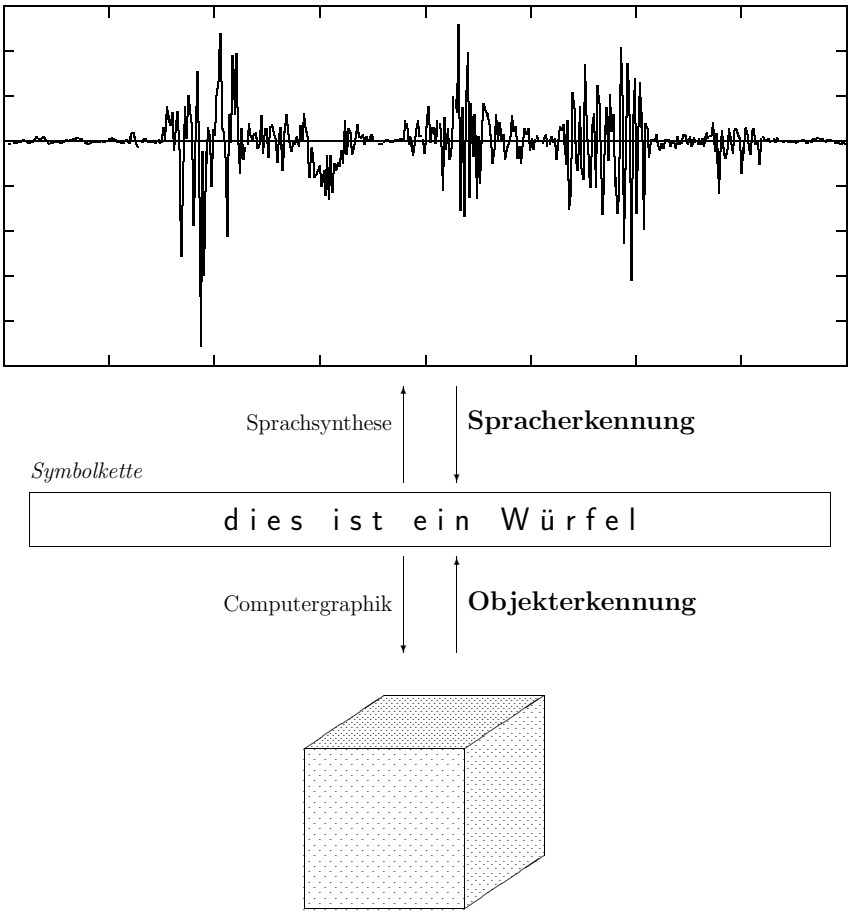


Bild 1.2.1: Mustererkennung leistet die automatische Transformation eines Signals in eine Symbolkette, z. B. bei der Sprach- und Objekterkennung

Neuigkeitsdetektion. Hierfür wird auf die Literaturangaben in Abschnitt 4.11 verwiesen. Bei soziologischen oder psychologischen Untersuchungen ist es dagegen möglich, dass Testpersonen Kennzeichen verschiedener Typen aufweisen. In diesen Fällen kann man entweder die Forderung nach Disjunktheit fallen lassen oder neben reinen Klassen Mischklassen einführen, die Muster mit den Kennzeichen mehrerer Klassen enthalten. Damit lässt sich auch in diesen Fällen die Forderung nach Disjunktheit der Klassen erfüllen.

Eine spezielle Form der Zerlegung von Ω ist die hierarchische Zerlegung, die in (1.2.8) mit enthalten ist. Man kann nämlich (1.2.8) als eine Zerlegung der Stufe 1 auffassen, bei der Teilmengen Ω_κ gebildet werden. In der Stufe 2 wird jede Teilmenge Ω_κ selbst wieder gemäß (1.2.8) in Teilmengen $\Omega_{\kappa\lambda}$ zerlegt. Dieser Prozess kann bei Bedarf noch über weitere Stufen fortgeführt werden. Ein Beispiel ist die Zerlegung der Schriftzeichen auf der Stufe 1 in Ziffern, Buchstaben und Sonderzeichen. Auf der Stufe 2 wird die Klasse der Ziffern zerlegt in zehn weitere Klassen, die den Ziffern 0 bis 9 entsprechen, und ähnlich werden die Klassen der Buchstaben und Sonderzeichen weiter zerlegt.

Definition 1.6 *Klassifikationsaufgaben können sowohl bei einfachen als auch bei komplexen Mustern auftreten.*

1) Bei der **Klassifikation** *von (einfachen) Mustern wird jedes Muster als ein Ganzes betrachtet und unabhängig von anderen Mustern genau einer Klasse Ω_κ von k möglichen Klassen*

Ω_λ, $\lambda = 1, \ldots, k$ *zugeordnet. Die* Rückweisung *eines Musters, also die Zuordnung zu einer* $(k+1)$*–ten Klasse* Ω_0*, ist zulässig.*

2) Ein (komplexes) Muster kann i. Allg. mehr als ein zu klassifizierendes Objekt enthalten. In diesem Falle ist eine Folge von Klassen $\mathbf{\Omega} = [{}^1\Omega, \ldots, {}^i\Omega, \ldots {}^N\Omega]$, ${}^i\Omega \in \Omega$ *zu bestimmen, bzw. Muster sind im* **Kontext** *anderer Muster zu klassifizieren.*

3) Zusätzlich zur Klassifikation kann auch eine **Lokalisation***, d. h. die Bestimmung der Translation* $\mathbf{t}$ *des Referenzpunktes eines Musters (relativ zu einem Referenzkoordinatensystem) sowie der Rotation* $\boldsymbol{R}$ *erforderlich sein. Das Ergebnis ist dann* $(\Omega_\kappa, \boldsymbol{t}_\kappa, \boldsymbol{R}_\kappa)$.

Beispiele für typische Klassifikationsaufgaben sind die Klassifikation von gedruckten Schriftzeichen einer oder weniger Schrifttypen, von isoliert gesprochenen Wörtern, von dreidimensionalen Objekten oder von Unternehmen auf der Basis von Kennzahlen–Mustern zum Zwecke der Erfolgsprognose. In den obigen Fällen sind die vorliegenden Muster relativ einfach und die Zahl der Klassen ist gering – typisch $k \leq 300$, bei Beleglesern reicht oft $k \simeq 14$. Es gibt jedoch auch Klassifikationsaufgaben, die wesentlich umfangreicher sind. Ein Beispiel ist die Klassifikation von Fingerabdrücken, d. h. die automatische Ermittlung der Identität eines unbekannten Abdrucks. Das Muster (der Fingerabdruck) hat eine viel kompliziertere Struktur als z. B. eine gedruckte Ziffer, die Zahl der Klassen, die der Zahl der in Frage kommenden Personen entspricht, ist um Größenordnungen gesteigert, da in Karteien i. Allg. die Abdrücke von mehreren Millionen Personen vorliegen. Hier liegt eine Aufgabe vor, die zwar auch auf eine Klassifikation hinausläuft, bei der aber zumindest der Übergang auch zur Analyse von Mustern vorhanden ist. Ein weiteres Beispiel ist die Worterkennung in Diktiersystemen, bei denen die Zahl der Klassen, d. h. der bekannten Wörter, $k \approx 10^4 - 10^5$ ist. Im Falle der Klassifikation einfacher Muster kommt es z. B. bei den Schriftzeichen darauf an, alle möglichen Realisierungen eines Zeichens mit bestimmter Bedeutung, beispielsweise der Ziffer 3, der gleichen Klasse zuzuordnen. Es kann passieren, dass eine solche Zuordnung nicht oder nicht genügend verlässlich möglich ist, wie es z. B. bei der Unterscheidung zwischen dem Buchstaben o und der Ziffer 0 oft der Fall sein kann; dann sollte das fragliche Muster der *Rückweisungsklasse* Ω_0 zugeordnet werden. Es wird nun noch kurz auf den Begriff des einfachen Musters eingegangen.

Definition 1.7 *Ein Muster wird als* **einfaches Muster** *betrachtet, wenn den Anwender nur der Klassenname interessiert und wenn es möglich ist, es als Ganzes genau einer Klasse zuzuweisen.*

Sicherlich kann die obige Definition nur als ein Anhaltspunkt und nicht als strenges Unterscheidungskriterium betrachtet werden. Bild 1.2.2 zeigt drei Beispiele für Muster, die im obigen Sinne als einfach zu bezeichnen sind. Obwohl gemäß dem Titel des Buches hier die Klassifikation von Mustern behandelt wird, ist es zweckmäßig, zur besseren Eingrenzung auch kurz zu definieren, was unter Analyse von Mustern zu verstehen ist.

Definition 1.8 *Bei der* **Analyse** *von (komplexen) Mustern wird jedem Muster* ${}^\varrho\boldsymbol{f}(\boldsymbol{x}) \in \Omega$ *eine individuelle symbolische Beschreibung* ${}^\varrho\mathcal{B}$ *zugeordnet. Sie besteht aus einem Netzwerk bzw. einer formalen Datenstruktur, die aus dem Muster berechnete Instanzen* I_j *von Konzepten* C_k *einer Wissensbasis bzw. eines Modells enthält*

$$ {}^\varrho\mathcal{B} = \langle I_j(C_k) \rangle \ . \tag{1.2.9}$$

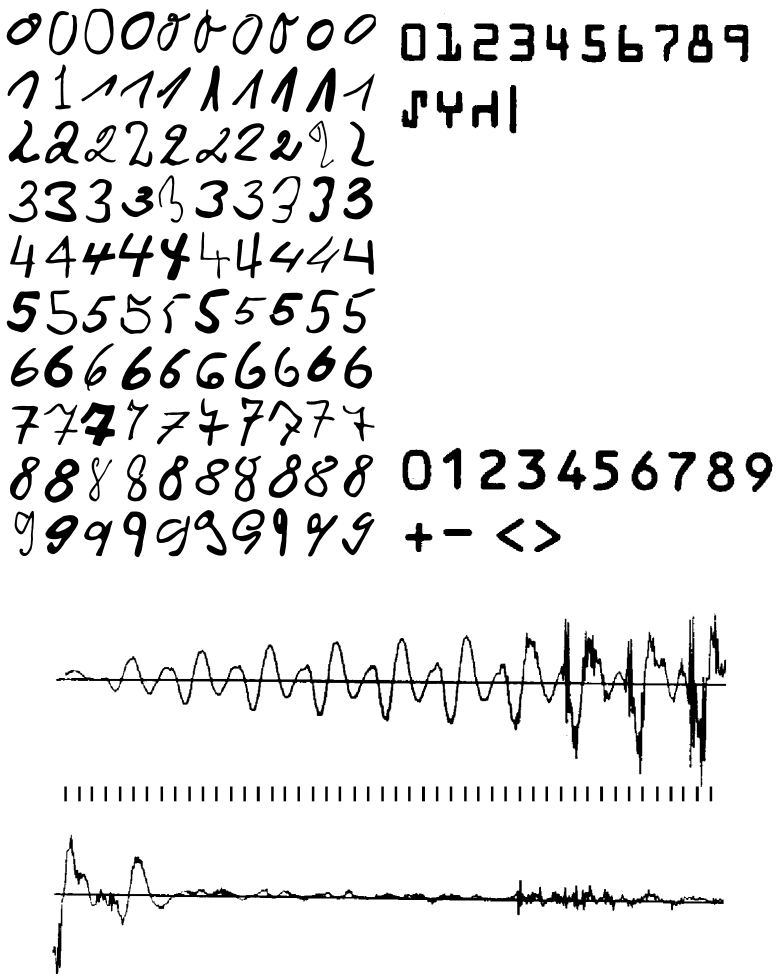


Bild 1.2.2: Drei Beispiele für einfache Muster. Oben links handgeschriebene Ziffern, oben rechts standardisierte Ziffern, unten der Spannungsverlauf am Mikrofonausgang für das Wort „mit“

Beispiele für Analyseaufgaben sind das automatische Verstehen zusammenhängend gesprochener Sprache, die Auswertung von Multispektralbildern in der Erdfernerkundung, die diagnostische Interpretation medizinischer Bilder oder die Ermittlung von Schaltelementen und Verbindungen in elektrischen Schaltplänen. In den obigen Fällen ist i. Allg. ein Klassenname nicht ausreichend, da er für den Anwender zu wenig aussagt, und es werden i. Allg. zahlreiche Einzelobjekte im Muster zu unterscheiden sein. Bei den Schaltplänen kann eine Klassenbezeichnung z. B. sein, dass es die Schaltung eines „Farbfernsehgerätes vom Typ ABC der Firma XYZ“ ist. Für Zwecke der Fertigung wird man dagegen alle Schaltelemente mit ihren genauen Bezeichnungen (beispielsweise „Widerstand $R10$ mit 2,7 kΩ“), alle Verbindungen zwischen Schaltelementen und, falls vorhanden, deren Bezeichnung (beispielsweise „Widerstand $R10$ ist über Verbindungsleitung $V7$ mit Basis von $T3$ verbunden“) sowie Endpunkte von Leitungen (beispielsweise Verbindungsleitung $V4$ endet am Punkt $P5$) ermitteln und in einer entsprechenden Datenstruktur speichern müssen. Natürlich ist nicht ausgeschlossen, dass man eine solche symbolische Beschreibung unter einem Klassennamen zusammenfasst.

Die Anwendung bestimmt, welche Information in einer Beschreibung enthalten sein soll.

Daher kann das Ziel der Analyse sehr unterschiedlich sein. Eine symbolische Beschreibung kann unter anderem folgende Information enthalten:

1. Eine ausführliche *symbolische Beschreibung* eines vorgelegten Musters. – Ein Beispiel sind die erwähnten Schaltpläne.
2. Eine Liste einiger *interessanter Objekte oder Ereignisse*, die in dem Muster enthalten sind. – Ein Beispiel ist die Ermittlung von Flugplätzen in einem Luftbild.
3. Eine Beschreibung von *Veränderungen*, die zwischen zeitlich aufeinander folgenden Aufnahmen eines Musters auftreten. – Ein Beispiel ist die Veränderung des Waldbestandes auf zwei in zeitlichem Abstand aufgenommenen Multispektralbildern der gleichen Landschaft.
4. Die *Klassifikation* eines komplexen Musters. – Ein Beispiel ist die Zuordnung einer der Diagnosen gesund, krank oder unklar zu einem Röntgenbild des Thorax.
5. Die aufgabenspezifische *Interpretation* eines Sachverhalts. – Ein Beispiel ist die diagnostische Interpretation medizinischer Bilder.
6. Die aufgabenspezifische *Reaktion* auf ein Muster. – Ein Beispiel ist die automatische Generierung einer Antwort auf eine (gesprochene) Frage.

Wegen der Vielfalt der Muster und der unterschiedlichen Ziele, die bei der Analyse verfolgt werden, ist es nahezu zwangsläufig, dass eine große Anzahl von Auswertemethoden entwickelt wurde und noch entwickelt wird. Als nächstes wird auf die Beschreibung eines Musters eingegangen.

Definition 1.9 *Unter der* **Beschreibung** *eines Musters wird die Zerlegung des Musters in einfachere Bestandteile und die Analyse von deren Beziehungen untereinander verstanden.*

Im Falle der Schaltpläne sind einfachere Bestandteile z. B. Widerstände und Transistoren, und ihre Beziehungen bestehen in elektrischen Verbindungen. Allgemein ist die Beschreibung eine andere Darstellung des Musters oder wichtiger Teile desselben, sodass bei der Analyse lediglich eine Transformation zwischen verschiedenen Repräsentationen der Information vorgenommen wird. Zwei wichtige Gründe für die Durchführung einer solchen Transformation sind:

1. Eine andere Repräsentation ist für die weitere Verarbeitung geeigneter.
2. In der neuen Repräsentation ist nur die für den Anwender wichtige Information enthalten.

Eine symbolische Beschreibung enthält die meiste Information, während die Zusammenfassung unter einem Klassennamen die komprimierteste Form ist. Als letztes wird noch der Begriff des komplexen Musters erläutert.

Definition 1.10 *Ein Muster wird als* **komplexes Muster** *betrachtet, wenn dem Anwender die Angabe eines Klassennamens nicht genügt oder wenn die Klassifikation als Ganzes in eine Klasse nicht möglich ist.*

In Bild 1.2.3 sind drei Beispiele für komplexe Muster angegeben. Ein Vergleich mit Bild 1.2.2 zeigt, dass diese – zumindest intuitiv – wesentlich komplizierter strukturiert sind. Natürlich sind einfache und komplexe Muster keine „reinen Typen“, vielmehr gibt es fließende Übergänge zwischen beiden. Es kann auch von der speziellen Anwendung, den benutzten Methoden und der Ansicht des Anwenders abhängen, ob ein bestimmtes Muster als einfach oder komplex bezeichnet wird. Trotz dieser Einschränkungen und Unschärfen scheint es nützlich, die Unterscheidung zwischen beiden als eine Möglichkeit der Strukturierung des sehr umfassenden Begriffs Muster zu verwenden.

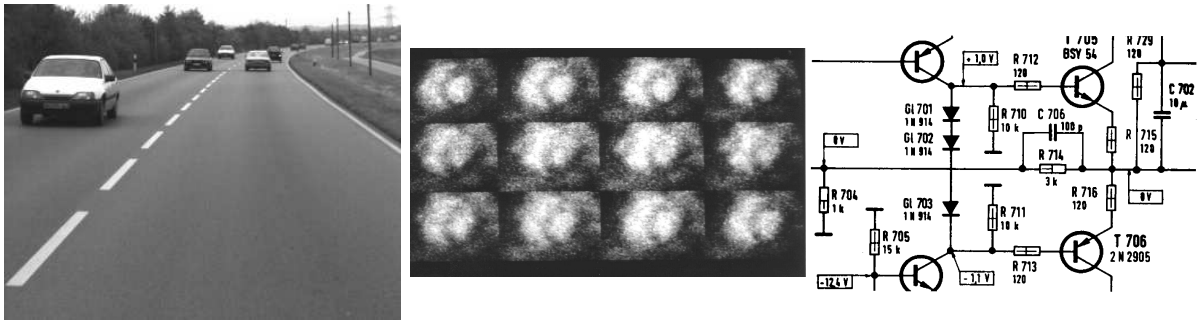


Bild 1.2.3: Drei Beispiele für komplexe Muster; links eine Verkehrsszene, in der Mitte eine szintigraphische Bildfolge vom Herz, rechts ein Ausschnitt aus einem Schaltplan

1.3 Grundsätzliche Vorgehensweise

Im Universum geschieht nichts, das nicht den Sinn eines bestimmten Maximums oder Minimums hat. (EULER)

Welchen Sinn, angenommen es gäbe einen, hätte denn ein Sinn. (GRASS)

Bei aller Verschiedenheit der Ansätze und Methoden der Mustererkennung zeichnet sich zunehmend eine einheitliche Vorgehensweise insofern ab, als immer mehr und größere Teilprobleme unter dem Aspekt der *Optimierung* geeignet gewählter Zielgrößen bzw. Gütefunktionen behandelt werden. Dies kommt in (1.3.7), (1.3.8) und (1.3.9) zum Ausdruck. Weiter liegen allen Systemen zur Klassifikation und Analyse von Mustern einige wenige gemeinsame Prinzipien zugrunde, die zunächst im Folgenden in sechs Postulaten zusammengefasst sind.

Postulat 1 *Zur Sammlung von Information über einen Problemkreis Ω steht eine repräsentative* **Stichprobe**

$$\omega = \{({}^1\boldsymbol{f}(\boldsymbol{x}), y_1), \ldots, ({}^\varrho\boldsymbol{f}(\boldsymbol{x}), y_\varrho), \ldots, ({}^N\boldsymbol{f}(\boldsymbol{x}), y_N)\} \subset \Omega \tag{1.3.1}$$

zur Verfügung, wobei ${}^\varrho\boldsymbol{f}(\boldsymbol{x})$ das ϱ–te Muster und y_ϱ Zusatzinformation über das Muster bedeutet.

Diese Forderung beruht auf der offensichtlichen Tatsache, dass man nicht ein konkretes System entwickeln kann, ohne gründliche Kenntnisse über die von dem System zu verarbeitenden Objekte zu haben. Es ist wichtig, dass die Stichprobe nur Muster aus dem interessierenden Problemkreis enthält, da man das System sonst für Fälle auslegt, die im konkreten Einsatz nie auftreten. Muster ${}^\varrho\boldsymbol{f}(\boldsymbol{x}) \notin \Omega$ bereiten natürlich dann kein Problem, wenn sie als solche gekennzeichnet sind. Weiterhin ist es wichtig, dass die Stichprobe repräsentativ ist, da die Schlüsse, die man aus ω zieht, nicht nur für alle ${}^\varrho\boldsymbol{f}(\boldsymbol{x}) \in \omega$ sondern auch für alle (oder doch zumindest möglichst viele) ${}^\varrho\boldsymbol{f}(\boldsymbol{x}) \in \Omega$ zutreffen sollen. Dieses ist das wichtige Problem der **Generalisierung** von Beobachtungen. Ob eine Stichprobe repräsentativ ist, ist i. Allg. schwierig zu entscheiden. Hinweise darauf geben jedoch die Untersuchung der Konfidenzintervalle von geschätzten Parametern in Abschnitt 4.10 und der Test der Systemleistung mit Mustern, die *nicht* in der Stichprobe ω enthalten sind. Wenn eine **repräsentative Stichprobe** zur Entwicklung eines Systems verwendet wurde und die Generalisierungsfähigkeit des Systems gegeben ist, so ist bei einem Test die Systemleistung nahezu unabhängig davon, ob die verarbeiteten Muster in der

Stichprobe enthalten waren oder nicht. Da man dieses vorher nicht sicher weiß, ist es notwendig, dass die zum Training eines Systems und zum Test des Systems verwendeten Stichproben *disjunkt* sind. Einen quantitativen Anhaltspunkt gibt auch die Kapazität eines Klassifikators in Abschnitt 4.10.

Einige Beispiele für *Zusatzinformation* y_ϱ sind

$$y_\varrho \in \{-1, 1\} , \tag{1.3.2}$$
$$y_\varrho \in \{1, \ldots, \kappa, \ldots, k\} , \tag{1.3.3}$$
$$y_\varrho : \{\Omega_{\varrho,1}, \ldots, \Omega_{\varrho,k_\varrho}\} \in {}^{\varrho}\boldsymbol{f}(\boldsymbol{x}) , \tag{1.3.4}$$
$$y_\varrho = \emptyset . \tag{1.3.5}$$

Die Form in (1.3.2) wird oft zur Charakterisierung des Zweiklassenproblems verwendet; man kann auch sagen, dass ein Muster entweder zu einer bestimmten Klasse gehört oder nicht. In (1.3.3) wird je Muster genau eine von k Klassen angegeben, zu der es gehört. Ein wesentlich allgemeineres Problem ist mit (1.3.4) charakterisiert; hier wird zu einem Muster lediglich angegeben, dass es Objekte aus einigen Klassen enthält, aber nicht, wieviele Objekte aus einer Klasse sind und in welcher räumlichen (oder zeitlichen) Anordnung. In (1.3.5) schließlich wird keinerlei Zusatzinformation über das Muster gegeben. Je nach verfügbarer Zusatzinformation ergeben sich unterschiedliche Probleme, die in Abschnitt 4.8.1 genauer charakterisiert werden. In den letzten beiden Fällen stellt sich das Problem, mit unvollständiger oder fehlender Information auf die Eigenschaften von Musterklassen zu schließen. Das grundsätzliche Hilfsmittel dafür ist der in Abschnitt 1.6.4 skizzierte EM-Algorithmus. Der Fall, dass $y_\varrho \in \mathbb{R}$ (mit $\mathbb{R}$ die Menge der reellen Zahlen) gegeben ist, bezeichnet das *Regressionsproblem*, das hier nicht weiter betrachtet wird.

Allgemein lässt sich sagen, dass der erforderliche Umfang N der Stichprobe ω nur von den statistischen Eigenschaften der Muster und der Art der zu schätzenden Parameter abhängt. Er ist dagegen völlig *unabhängig* von den Kosten, die die Aufnahme eines Musters verursacht. Allerdings hat der zum Teil erhebliche Aufwand an Geld und Zeit, den die Sammlung einer großen Stichprobe verursacht, meistens zur Folge, dass der Stichprobenumfang eher zu klein als zu groß gewählt wird. Für Systeme zur Klassifikation von Mustern sind außer dem obigen Postulat noch die beiden folgenden wichtig.

Postulat 2 *Ein (einfaches) Muster besitzt* **Merkmale***, die für seine Zugehörigkeit zu einer Klasse charakteristisch sind.*

Postulat 3 *Die Merkmale bilden für Muster einer Klasse einen einigermaßen kompakten Bereich im Merkmalsraum. Die von Merkmalen verschiedener Klassen eingenommenen Bereiche sind einigermaßen getrennt. Dieses ist die wichtige* **Kompaktheitshypothese** *für Merkmale.*

Das zentrale und allgemein noch ungelöste Problem der Klassifikation besteht darin, solche Merkmale systematisch zu finden, die Postulat 3 genügen. Damit ist hier ein Algorithmus gemeint, der nach Vorgabe einer Stichprobe und eines Maßes für die Leistungsfähigkeit des Systems Merkmale erzeugt, die dieses Maß maximieren (oder minimieren). Trotzdem konnte empirisch nachgewiesen werden, dass es zumindest für bestimmte Problemkreise geeignete kompakte Merkmale gibt. Bereiche im Merkmalsraum mit unterschiedlicher Kompaktheit sind in Bild 1.3.1 angedeutet.

Beispiele für die Verteilung von Merkmalen zeigt Bild 1.3.2 für drei Typen von Mustern. Dargestellt sind jeweils die Beträge der Koeffizienten $F_{0,1}$, $F_{1,0}$ der *diskreten* FOURIER-*Transformation*, die in Abschnitt 2.3.3 definiert ist. Alle Beträge sind so normiert, dass der

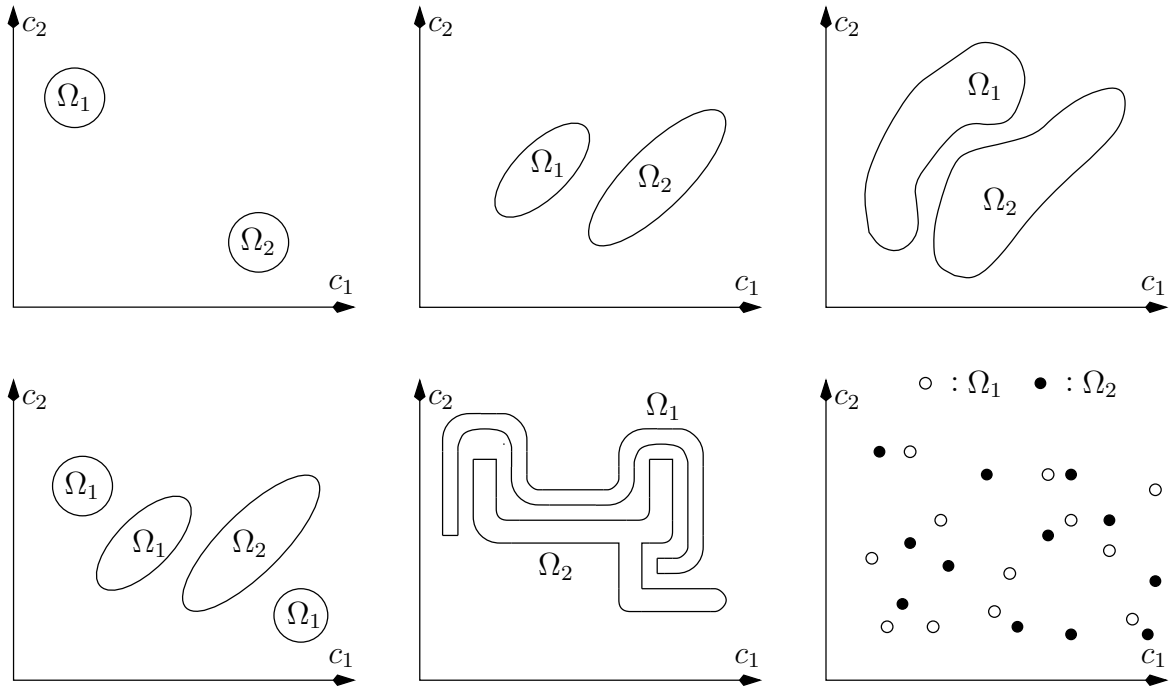


Bild 1.3.1: Beispiele für zunehmend weniger kompakte Bereiche im Merkmalsraum

grßte Wert von $|F_{0,1}|$ gleich Eins ist. Die Bilder der ersten Zeile gehen durch eine lineare Transformation, nämlich Rotation des Ausgangsbildes „Lena“ in der Bildebene um den Winkel $\alpha = 2\pi/101$, auseinander hervor. Die beiden Merkmale, dargestellt links in der letzten Zeile des Bildes, durchlaufen eine relativ übersichtliche und einfache Trajektorie. Die Bilder der zweiten Zeile sind das Objekt Nr. 03 der Stichprobe COIL-20. Auch sie werden durch eine Rotation erzeugt, aber eine Rotation um $\alpha = 2\pi/72$ des dreidimensionalen Objekts aus der Bildebene heraus, was allmählich zur völligen Veränderung der Ansichten führt. Die Trajektorie wird recht unübersichtlich, wie aus dem Bild in der Mitte der letzten Zeile hervorgeht. Die Bilder der dritten Zeile sind die ersten sechs Ziffern der Klasse „zwei“ der MNIST-Teststichprobe. Bei den beiden Merkmalen, die für die ersten 100 Ziffern berechnet wurden und die im Bild rechts in der letzten Zeile dargestellt sind, hat es keinen Sinn mehr, eine Trajektorie einzuzeichnen.

Das eigentliche Klassifikationsproblem, d. h. die Abgrenzung der zu den Klassen gehörenden Bereiche und die Zuordnung eines neuen Musters zu einem dieser Bereiche, ist dagegen prinzipiell gelöst. Da die Merkmale i. Allg. als Komponenten c_ν eines Merkmalsvektors $\boldsymbol{c}$ aufgefasst werden, bedeutet Klassifikation eines neuen Musters ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x})$ also eine Abbildung

$$ {}^{\varrho}\boldsymbol{c} \rightarrow \kappa \in \{1, \ldots, k\} \quad \text{oder} \quad {}^{\varrho}\boldsymbol{c} \rightarrow \kappa \in \{0, 1, \ldots, k\} \tag{1.3.6} $$

des aus ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x})$ extrahierten Merkmalsvektors ${}^{\varrho}\boldsymbol{c}$. Im Prinzip hat ein System zur Klassifikation von Mustern die in Bild 1.3.3 gezeigte **hierarchische Struktur**. Es besteht aus einigen Systemkomponenten oder Moduln, die bestimmte Verarbeitungsschritte oder Transformationen ausführen. Die Ausgangsgröße des Moduls i wird dabei die Eingangsgröße des nachfolgenden Moduls $(i + 1)$. Eine derartige Struktur ist relativ leicht überschaubar, es ist naheliegend, jeden einzelnen Modul für sich zu realisieren und zu optimieren, und es gibt erfahrungsgemäß wichtige praktische Probleme, bei denen diese Systemstruktur zu befriedigenden Lösungen führt.

Ein sehr allgemeiner und leistungsfähiger Ansatz zur Klassifikation (und Lokalisation) von Mustern besteht darin, den Entscheidungen für die eine oder andere Klasse *Kosten* zuzuordnen

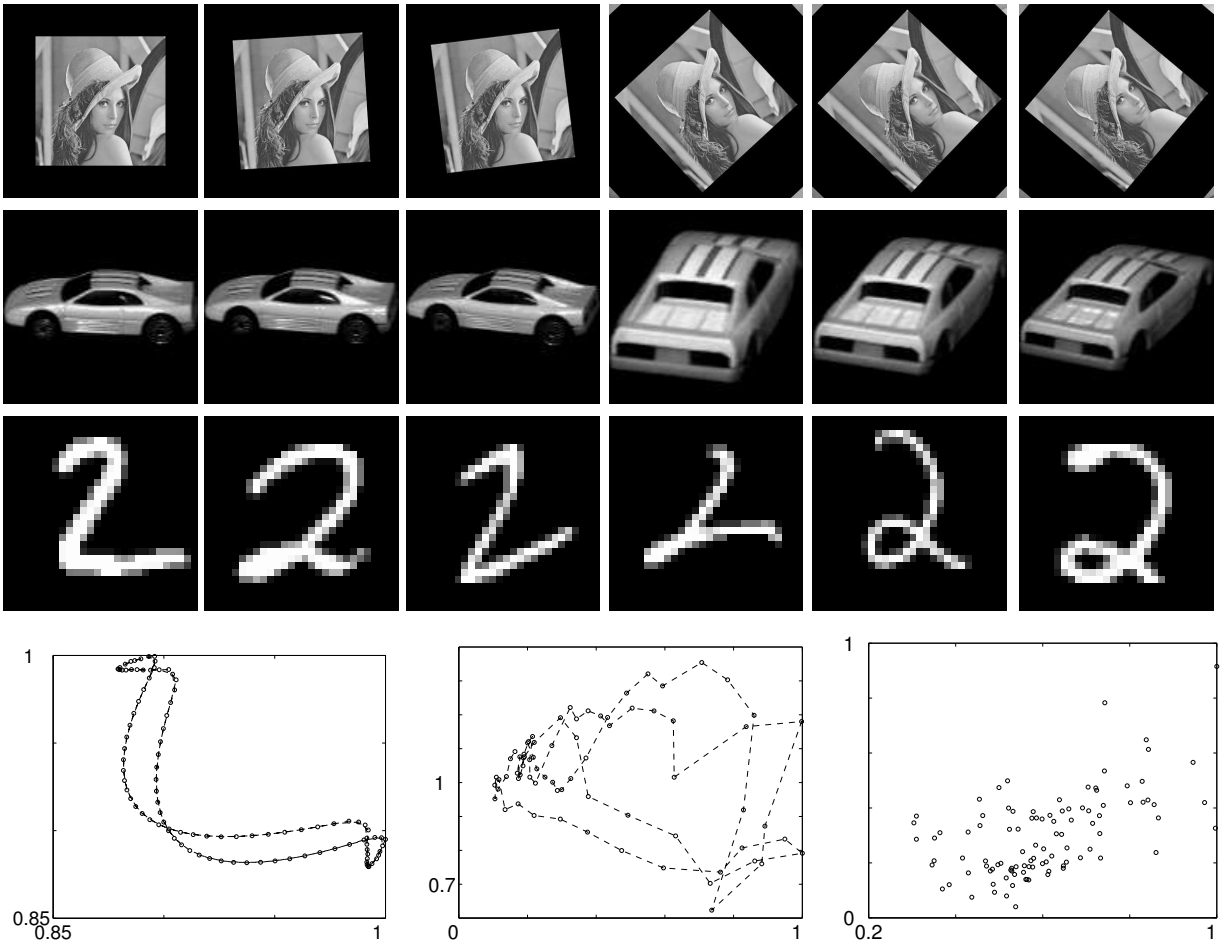


Bild 1.3.2: Die Beträge der Koeffizienten $F_{0,1}$, $F_{1,0}$ der DFT für einige Muster; Einzelheiten gehen aus dem Text hervor

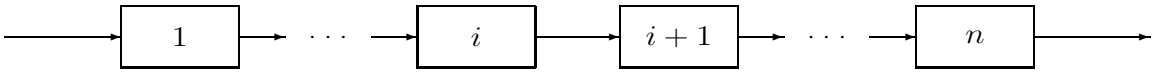


Bild 1.3.3: Ein hierarchisch strukturiertes System

und nach der Entscheidungsregel δ bzw. dem Klassifikator zu suchen, der die mittleren Kosten V minimiert,

$$\delta^* = \underset{\{\delta\}}{\operatorname{argmin}} V(\delta) \; . \tag{1.3.7}$$

Die Klassifikation von Mustern ist also ein *Optimierungsproblem*. Dabei sind eine Vielzahl von Einzelheiten festzulegen, die zu einer Vielzahl möglicher Lösungen des Klassifikationsproblems führen. Auf Einzelheiten wird in Kapitel 4 eingegangen.

Ein System zur Analyse von Mustern basiert neben Postulat 1 noch auf den Postulaten 4 und 5.

Postulat 4 *Ein (komplexes) Muster besitzt* **einfachere Bestandteile**, *die untereinander bestimmte Beziehungen haben. Das Muster lässt sich in diese Bestandteile zerlegen.*

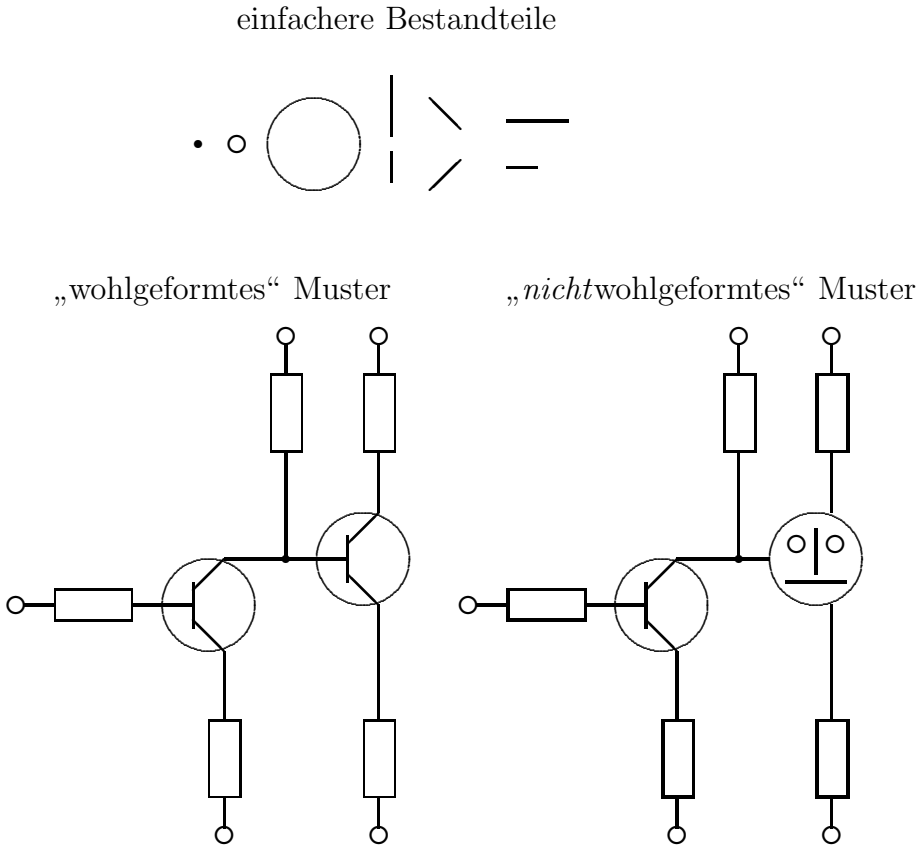


Bild 1.3.4: Aus den gleichen einfacheren Bestandteilen lassen sich, relativ zu einem Problemkreis, sinnvolle und sinnlose Muster konstruieren

Postulat 5 *Ein (komplexes) Muster aus einem Problemkreis hat eine bestimmte* **Struktur**. *Das bedeutet, dass nicht jede beliebige Anordnung einfacherer Bestandteile ein Muster* $^{\varrho}\boldsymbol{f}(\boldsymbol{x}) \in \Omega$ *ergibt und dass weiter sich viele Muster mit relativ wenigen einfacheren Bestandteilen darstellen lassen.*

Das Problem, geeignete einfachere Bestandteile zu finden, ist ebenfalls in allgemeiner Form ungelöst. Jedoch liegen auch hier experimentelle Ergebnisse und Erfahrungen vor, die zeigen, dass es solche einfacheren Bestandteile zumindest für bestimmte Problemkreise gibt. Ein Beispiel dafür sind die Laute oder Phoneme als einfachere Bestandteile der Sprache. Für Formalismen zur Erfassung struktureller Eigenschaften und zur Analyse von Mustern liegen eine Reihe von Ansätzen und Ergebnissen vor. Bild 1.3.4 zeigt, dass es offensichtliche strukturelle Einschränkungen in der Anordnung einfacherer Bestandteile gibt, da nicht jede Anordnung ein gültiges oder sinnvolles Muster ergibt. Ein analoges Beispiel aus der Spracherkennung sind die Sätze „nachts ist es kälter als am Tage“ und „nachts ist es kälter als draußen“.

Für ein System zur Analyse von Mustern ist es vielfach unzweckmäßig, wenn Transformationen in der festen und unveränderlichen Reihenfolge des hierarchischen Systems ausgeführt werden. Eine flexiblere Struktur zeigt Bild 1.3.5 mit der **datenbankorientierten Struktur**, die zwar auch eine Reihe von Moduln zur Ausführung von Transformationen enthält, jedoch keine Reihenfolge der Aktivierung dieser Moduln angibt. Die Moduln sind über eine gemeinsame Datenbank, die Zwischenergebnisse der bisherigen Verarbeitung enthält, gekoppelt. Ein spezieller Modul, der Kontrollmodul, entscheidet für jedes zu verarbeitende Muster, welche Transformati-

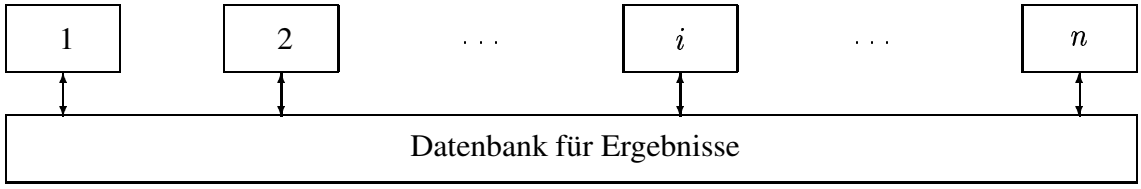


Bild 1.3.5: Ein datenbankorientiertes System

on jeweils auszuführen ist. Damit ist i. Allg. die Reihenfolge der Verarbeitungsschritte abhängig von dem zu verarbeitenden Muster. Diese als datenbankorientiertes System bezeichnete Struktur enthält das hierarchische System als Spezialfall. Wenn nämlich der Kontrollmodul für alle (oder fast alle) Muster die gleiche Folge von Verarbeitungsschritten auswählt, dann kann man auf ihn verzichten und diese Schritte explizit in einer Struktur gemäß Bild 1.3.3 festhalten.

Die Ansätze zur Lösung des Problems der Analyse sind äußerst heterogen, was z. T. daran liegt, dass die Elemente der Beschreibung $\mathcal{B}$ je nach Anwendung sehr verschieden sind. Um jedoch einen ersten Eindruck auch von dieser Vorgehensweise zu geben, wird ein möglicher Ansatz kurz skizziert. Man repräsentiert a priori Wissen über den Problemkreis in einem Modell $\mathcal{M}$, das als ein Netzwerk (oder Graph) von Konzepten C aufgebaut ist. In Frage kommen z. B. semantische Netze oder BAYES-Netze. Ein Konzept ist die rechnerinterne Darstellung eines Begriffs, Objekts, Sachverhalts oder Ereignisses. Auch das Ziel der Analyse wird als Konzept, nämlich als Zielkonzept C_g, repräsentiert. Für ein zu analysierendes Muster wird eine initiale Segmentierung $\mathcal{A}$ berechnet, z. B. eine Segmentierung in Linien. Es gibt eine Schnittstelle zwischen der initialen Segmentierung und dem Modell. Wenn also z. B. die initiale Segmentierung Linien liefert, muss es im Modell Konzepte geben, die Linien repräsentieren. Als Ergebnis der Analyse werden Instanzen von Konzepten berechnet. Während das Konzept den allgemeinen Begriff repräsentiert, z. B. ein Haus, das Attribute wie Breite, Länge und Dachfarbe hat, repräsentiert eine Instanz einen speziellen im Muster gefundenen Begriff, z. B. ein Haus an bestimmter Stelle mit bestimmter Breite, Länge und Dachfarbe. Jedes Konzept, auch das Zielkonzept, hat eine zugeordnete Bewertungsfunktion G, mit der die Güte oder Zuverlässigkeit einer Instanz des Konzepts berechnet werden kann. Die Analyse läuft dann darauf hinaus, die bestbewertete Instanz des Zielkonzepts zu berechnen, wenn ein bestimmtes Modell und eine bestimmte initiale Segmentierung gegeben sind, also

$$\begin{aligned} \mathcal{B} &= I^*(C_g) \\ &= \underset{\{I(C_g)\}}{\operatorname{argmin}}\, G\left(I(C_g)|\mathcal{M},\mathcal{A}\right) \ . \end{aligned} \tag{1.3.8}$$

Die Analyse von Mustern ist also ein *Optimierungsproblem*. Damit ist auch eine formalere Definition des Begriffs *„symbolische Beschreibung“* gegeben. Da $I^*(C_g)$ selbst von vorangehenden Instanzen abhängt, ergibt sich ein Netzwerk von Konzepten (1.2.9). Auf weitere Einzelheiten wird im Rahmen dieses Buches nicht eingegangen.

Eine weitere Strukturvariante, die in Bild 1.3.6 angedeutet ist, ergibt sich im Falle des aktiven Eingreifens eines Systems zur Mustererkennung in den Prozess der Aufnahme von Mustern, indem *Aktionen* zur Wahl der Kameraparameter, der Parameter von Aktoren und/oder der Beleuchtungsparameter bestimmt werden. Dieses wird insbesondere für bestimmte Anwendungen durch **aktives Sehen** als wichtiger Ansatz verfolgt. In diesem Fall sind die Parameter der Aufnahmebedingungen ${}^{g}\Xi_{\sigma}$ in (1.2.4) für jede Aufnahme eines Musters durch eine Steuerungs-

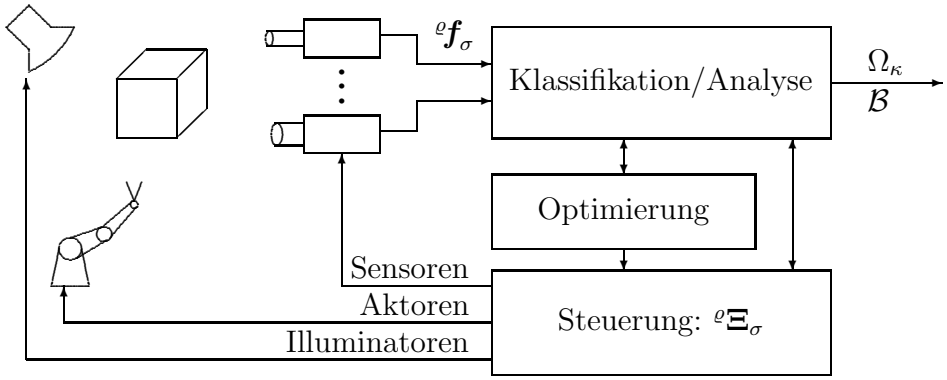


Bild 1.3.6: Ein aktives System zur Verarbeitung (Klassifikation, Analyse) von Mustern

einheit bzw. eine Aktionsauswahl *individuell* zu bestimmen. Hier betrachten wir einen zeitabhängigen Parametervektor $\boldsymbol{\Xi}_t$, der so bestimmt werden soll, dass ein (kontinuierlicher oder diskreter) zeitabhängiger Zustandsvektor $\boldsymbol{q}_t$, der die interessierenden Größen wie Lokalisation oder Klasse enthält, möglichst gut geschätzt werden kann. Ein Ansatz besteht darin, die Transinformation H zu berechnen, die ein Merkmalsvektor (oder eine Beobachtung) $\boldsymbol{c}_t$ zur Zeit t über den Zustand $\boldsymbol{q}_t$ enthält, unter der Voraussetzung dass der Parametervektor $\boldsymbol{\Xi}_t$ verwendet wurde. Der optimale Parametervektor ist dann der, der die Transinformation maximiert

$$\boldsymbol{\Xi}_t^* = \underset{\boldsymbol{\Xi}_t}{\operatorname{argmax}}\ H(\boldsymbol{q}_t, \boldsymbol{c}_t | \boldsymbol{\Xi}_t)\ . \tag{1.3.9}$$

Auch aktives Sehen lässt sich also als Optimierungsproblem formulieren.

Auch aus der obigen Diskussion wird klar, dass Klassifikation und Analyse keine disjunkten Bereiche sind, sondern vielmehr Gemeinsamkeiten und Überschneidungen bestehen. Hier gilt analog das bereits oben Gesagte, dass nämlich die Unterscheidung beider Begriffe eine gewisse Strukturierung der methodischen Vorgehensweise im relativ weiten Felde der Mustererkennung ermöglicht. Zu den Gemeinsamkeiten gehört z. B., dass bestimmte Merkmale auch als einfache Bestandteile aufgefasst werden können. Wenn man beispielsweise in einem Schriftzeichen Linienanfänge, Kreuzungen, senkrechte Striche und ähnliches ermittelt, so lassen sich diese ohne weiteres als einfachere Bestandteile des Schriftzeichens auffassen, andererseits aber auch ohne weiteres den Komponenten eines Merkmalsvektors zuordnen, indem man eine bestimmte Komponente Eins setzt, wenn eine Kreuzung vorhanden ist, und sonst Null setzt. Im Allgemeinen werden Merkmale meistens durch Zahlenwerte gekennzeichnet und einfachere Bestandteile durch Symbole, Listen von Attribut–Wert–Paaren oder allgemeiner durch formale Datenstrukturen, wobei das obige Beispiel deutlich macht, dass es durchaus Überschneidungen gibt. Ähnlich werden bei Klassifikationssystemen überwiegend – und bei den zur Zeit kommerziell erhältlichen Geräten sogar ausschließlich – numerische Rechnungen ausgeführt, während bei Analysesystemen die Manipulation von Symbolen eine große Bedeutung hat.

Im Zusammenhang mit der Definition der Musterklassen war unter anderem gesagt worden, dass Muster einer Klasse einander ähnlich sein sollen. Das nächste und letzte Postulat gibt die Grundlage für die Beurteilung von **Ähnlichkeit**.

Postulat 6 *Zwei Muster sind ähnlich, wenn ihre Merkmale oder ihre einfacheren Bestandteile sich nur wenig unterscheiden.*

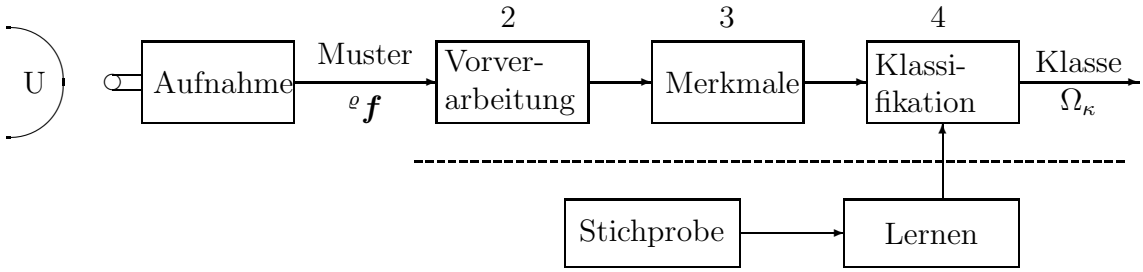


Bild 1.4.1: Die wesentlichen Moduln eines Systems zur Klassifikation von Mustern

Zwar mag Postulat 6 in dieser Form selbstverständlich sein, aber es ist die Basis aller Ansätze zur automatischen Bildung von Klassen einfacher Muster und auch von Mengen komplexer Muster mit ähnlichen Eigenschaften. Bei *numerischen* Merkmalen lassen sich Unterschiede durch Metriken und andere Abstandsmaße definieren. „Wenig unterscheiden“ heißt dann, dass der Wert des Abstandsmaßes unterhalb einer Schwelle bleibt. Ähnlich lässt sich bei Beschreibungen verfahren. Ein Beispiel sind die beiden Beschreibungen „das Objekt ist 4,8 m lang, hat die Farbe gelb, und hat 4 Räder“ und „das Objekt ist 4,8 m lang, hat die Farbe grün, und hat 4 Räder“. Sie unterscheiden sich nur in einem einfacheren Bestandteil, nämlich der Farbe, und können daher durchaus als ähnlich bezeichnet werden. Allerdings unterscheidet sich die Beschreibung „das Objekt ist 4,8 m lang, hat die Farbe gelb, und hat 4 Ruder“ von der ersteren auch nur in einem Punkt. Trotzdem wird man das letzte Objekt intuitiv als weniger ähnlich betrachten. Dieses lässt sich durch verschiedene *Gewichtung* der Unterschiede in den einfacheren Bestandteilen der Beschreibung berücksichtigen. Eine solche Gewichtung ist i. Allg. auch bei numerischen Merkmalen nützlich. Postulat 6 ist auch eine Ergänzung der Postulate 2 und 4: Merkmale oder einfachere Bestandteile müssen, wenn sie nützlich sein sollen, so gewählt werden, dass den Anwender interessierende Ähnlichkeiten in ihnen zum Ausdruck kommen.

1.4 Thematik des Buches

Schwer ist das Verständnis des Schönen, und das Verständnis der Namen ist keine leichte Aufgabe. (PLATON)

Wenn ich beim Malen meiner Bilder nicht weiß was sie bedeuten, so heißt das nicht, dass sie keine Bedeutung haben. (DALÍ)

In diesem Buch wird, wie auch der Titel festlegt, ausschließlich das Teilgebiet der *Klassifikation* von Mustern behandelt. Weiterhin werden ausschließlich *digitale Verarbeitungsverfahren* berücksichtigt. Als primäre Quelle von Information über die zu klassifizierenden Muster werden *Sensordaten* angesehen, d. h. Messwerte von physikalischen Größen. Für die Verarbeitung wird von der hierarchischen Systemstruktur ausgegangen, deren Prinzip Bild 1.3.3 zeigt und die in Bild 1.4.1 unter Angabe der wesentlichen Moduln nochmals wiederholt wird. Die Zahlenangaben bei den Moduln verweisen auf die Kapitel des Buches, in denen diese behandelt werden.

Gemäß Bild 1.4.1 wird ein zu klassifizierendes Muster $^{\varrho}\boldsymbol{f}(\boldsymbol{x}) \in \Omega \subset U$ zunächst aufgenommen, d. h. für Zwecke der weiteren Verarbeitung mit einem Rechner digitalisiert. Aufnahmegeräte werden hier nicht behandelt, da es dabei um Mess– und Sensorprobleme geht, die nicht im Vordergrund dieses Buches stehen. Wie bereits erwähnt, sind Bilder, Sprache und

Geräusche praktisch besonders wichtige Beispiele für Muster. Bei Bildern muss das Aufnahmegerät i. Allg. eine physikalische Größe, speziell z. B. die Lichtintensität, unter Umständen in verschiedenen Spektralkanälen, in eine elektrische Spannung umwandeln und bei Sprache oder Geräuschen den Schalldruck. Dafür eignen sich unter anderem Fotodioden, Fernsehkameras und Mikrofone. Das Problem, ein Muster $\boldsymbol{f}(\boldsymbol{x})$ mit kontinuierlichem Wertebereich für $\boldsymbol{f}$ und $\boldsymbol{x}$ digital – also mit einem diskreten Wertebereich für $\boldsymbol{f}$ und $\boldsymbol{x}$ – darzustellen, wird in Abschnitt 2.1 behandelt.

Nach der Aufnahme wird das Muster vorverarbeitet. Dabei soll vor allem die Qualität des Musters in der Hinsicht verbessert werden, dass die nachfolgende Verarbeitung erleichtert (Reduzierung des Aufwandes) und/oder die Klassifikationsleistung erhöht wird (Verbesserung der Leistung). Anschließend werden Merkmale extrahiert, deren Existenz und Eigenschaften mit den Postulaten 2 und 3 vorausgesetzt wurde. Wie im vorigen Abschnitt angedeutet wurde, können die Merkmale Zahlenwerte oder Symbole sein. Im ersten Falle werden die Zahlen den Komponenten eines Merkmalsvektors zugeordnet, im letzteren wird eine Kette von Symbolen gebildet. Die Merkmale werden dann klassifiziert, d. h. die in (1.3.6) angedeutete Abbildung ausgeführt. Je nach Typ der Merkmale kommen dafür numerische oder syntaktische Klassifikatoren in Frage.

Um Muster zu klassifizieren, müssen dem Klassifikator die Bereiche der Klassen bekannt sein. Diese werden in einer Lern– oder Trainingsphase mit Hilfe der Stichprobe ω ermittelt. In Bild 1.4.1 ist die Lernphase mit angedeutet. Auf Lernalgorithmen wird in Kapitel 4, in dem die entsprechenden Klassifikatoren behandelt werden, eingegangen. Die Leistungsfähigkeit eines Klassifikationssystems wird in der Regel zunächst durch Simulation des Systems am Digitalrechner ermittelt. Es wird hier angenommen, dass die Programmierung der verwendeten Algorithmen keine Probleme bereitet und daher übergangen werden kann. Bei zufriedenstellender Leistung des Systems und entsprechendem Bedarf kann dann eine Realisierung des Gesamtsystems durch spezielle Hardwarekomponenten erfolgen. Auch darauf wird im Rahmen dieses Buches nicht eingegangen.

Die hier als Basis verwendete einfache Systemstruktur in Bild 1.4.1 bzw. Bild 1.3.3 deckt eine Vielzahl praktisch wichtiger und theoretisch anspruchsvoller Klassifikationsaufgaben ab. In Bild 1.3.5 und Bild 1.3.6 werden mögliche Alternativen und Erweiterungen angedeutet, die schon aus Platzgründen nicht weiter behandelt werden.

1.5 Klassifikationsprobleme

Diejenigen, die BAYES-Methoden zurückweisen, zeigen nur – durch die benutzten Argumente – ihre Unwissenheit darüber, was BAYES-Methoden sind. (JAYNES)

Hat man dann die Teile in der Hand, fehlt leider nur das geistige Band. (GOETHE)

Nach einem einführenden Beispiel werden schematisiert einige Typen von Klassifikationsproblemen vorgestellt, die durch wachsende Komplexität charakterisiert sind. Ihr Lösungsprinzip beruht stets auf der Minimierung einer geeigneten Kostenfunktion, wie in (1.3.7) eingeführt. Zwei wichtige Beispiele für Kostenfunktionen sind in (4.1.10), S. 309, und (4.3.2), S. 361, angegeben.

Zwei Behälter

An den Anfang wird ein vereinfachtes Problem gestellt. Gegeben seien zwei äußerlich gleiche Behälter Ω_1 und Ω_2. Der Behälter Ω_1 enthält 100 rote und 900 blaue Kugeln, der Behälter Ω_2 enthält 900 rote und 100 blaue Kugeln. Einer der Behälter wird zufällig herausgegriffen. Offensichtlich ist die **a priori Wahrscheinlichkeit** dafür, den Behälter Ω_1 zu haben, gleich 0.5. Nun wird aus dem gewählten Behälter eine Kugel entnommen, d. h. es wird eine *Beobachtung* gemacht bzw. der *Wert* eines Merkmals bestimmt. Die Kugel sei blau, und die Frage ist, wie groß nun die Wahrscheinlichkeit für den Behälter Ω_1 ist. Gesucht ist also die **a posteriori Wahrscheinlichkeit** dafür, dass eine bestimmte Alternative vorliegt (Ω_1 oder Ω_2), nachdem eine bestimmte Beobachtung gemacht wurde (eine blaue Kugel). Aus bekannten Beziehungen über bedingte Wahrscheinlichkeiten, insbesondere aus der BAYES-**Regel** (4.1.3), S. 306, folgt

$$\begin{aligned} P(\Omega_1|b) &= \frac{P(b|\Omega_1)P(\Omega_1)}{P(b)} \\ &= \frac{P(b|\Omega_1)P(\Omega_1)}{P(b|\Omega_1)P(\Omega_1) + P(b|\Omega_2)P(\Omega_2)} \\ &= 0,9\ . \end{aligned} \tag{1.5.1}$$

Wir haben damit eine Gleichung, die die a priori Wahrscheinlichkeit $P(\Omega_1)$ der Urne *vor* einer Beobachtung in die a posteriori Wahrscheinlichkeit $P(\Omega_1|b)$ *nach* Beobachtung einer blauen Kugel transformiert.

Es wird nun eine zweite Kugel entnommen, die ebenfalls blau ist. Wie groß ist danach die Wahrscheinlichkeit für den Behälter Ω_1? Gesucht wird also die Wahrscheinlichkeit $P(\Omega_1|b,b)$, die sich ergibt zu

$$\begin{aligned} P(\Omega_1|b,b) &= \frac{P(b,b|\Omega_1)P(\Omega_1)}{P(b,b|\Omega_1)P(\Omega_1) + P(b,b|\Omega_2)P(\Omega_2)}\ , \\ &= 0,988\ . \end{aligned} \tag{1.5.2}$$

Daraus ist der *wichtige Schluss* zu ziehen, dass man *unsichere* Einzelbeobachtungen (die Farbe einer Kugel gibt keine Sicherheit über den Behälter) so kombinieren kann, dass das Gesamtereignis (gewählter Behälter) *immer sicherer* wird. Weiter ist festzustellen, dass zur Berechnung der a posteriori Wahrscheinlichkeiten die **bedingte Wahrscheinlichkeit** der Kugelfarbe $P(b|\Omega_\kappa)$ erforderlich ist.

Die Verallgemeinerungen für das Problem der Klassifikation von Mustern sind:

- Ein Behälter $\Rightarrow$ eine Musterklasse $\Omega_\kappa \Rightarrow$ eine Folge von Klassen $\boldsymbol{\Omega}$;
- eine Kugel $\Rightarrow$ ein Merkmal $c_\nu \Rightarrow$ ein Vektor $\boldsymbol{c}$ von Merkmalen bzw. eine Folge von Attributen;
- die Farbe einer Kugel $\Rightarrow$ der Wert eines Merkmals $\Rightarrow$ ein Folge von beobachteten Werten;
- berechne $P(\Omega_1|b,b) \Rightarrow$ berechne $P(\Omega_\kappa|\boldsymbol{c})$, $\kappa = 1,\ldots,k$ und wähle die Klasse (den Behälter) mit größter a posteriori Wahrscheinlichkeit.

In Abschnitt 4.1 wird gezeigt, dass die letztere intuitive Strategie, sich für die Alternative mit größter a posteriori Wahrscheinlichkeit zu entscheiden, sogar optimal in dem Sinne ist, dass sie die Fehlerwahrscheinlichkeit minimiert.

Wie erwähnt ist die Basis dieser Vorgehensweise die Berechnung der a posteriori Wahrscheinlichkeiten aller alternativen Entscheidungen bzw. Klassen. Dieses erfordert ein *stocha-*

stisches Modell der Muster des Problemkreises, das i. Allg. eine bestimmte Struktur und eine Menge freier Parameter hat. Es muss folgende Anforderungen erfüllen:

- Die Struktur des Modells muss für die Muster des Problemkreises adäquat sein, d. h. einen für die Klassifikation relevanten Ausschnitt der Umwelt hinreichend genau approximieren.
- Die Parameter des Modells müssen weitgehend automatisch trainierbar (oder schätzbar, lernbar) sein.
- Das Modell muss eine effiziente Berechnung der a posteriori Wahrscheinlichkeiten zulassen.

Beispiele für stochastische Modelle werden in Kapitel 4 vorgestellt.

Ein Muster als Ganzes

Die Klassifikation eines einzelnen Merkmalsvektors zeigt Bild 1.5.1. Beispiele dafür, wie gedruckte Schriftzeichen oder isoliert gesprochene Wörter, sind in Bild 1.2.2 gezeigt. Bei diesem „klassischen“ Problem geht man von der Voraussetzung aus, dass durch Aufnahmetechnik und Vorverarbeitung genau ein interessierendes Objekt (im Bild ist das ein Schriftzeichen) in der Aufnahme enthalten ist. Jede Klasse wird durch ein geeignetes *Modell* repräsentiert (im Bild ist das durch einen Prototypen $\boldsymbol{f}_\lambda$ angedeutet), wobei ein Prototyp hier einfach aus den Abtastwerten eines typischen Musters der Klasse durch Berechnung eines Merkmalsvektors gebildet wird. Andere und allgemeinere Ansätze zur Berechnung von Repräsentationen einer Klasse werden in Kapitel 4 behandelt. Die Berechnung von Merkmalen ist Thema von Kapitel 3. Ein neues Muster ${}^{\varrho}\boldsymbol{f}$ wird zunächst vorverarbeitet, z. B. in der Größe normiert. Verfahren zur Vorverarbeitung werden in Kapitel 2 vorgestellt. Für das vorverarbeitete neue Muster wird ebenfalls ein Merkmalsvektor berechnet. Ein Vergleich der beiden Merkmalsvektoren, der z. B. in der Berechnung des EUKLID-Abstands bestehen kann, ergibt Aufschluss, ob das neue Muster zu der betrachteten Klasse gehören kann oder nicht. Theoretisch fundierte Ansätze zur Klassifikation werden in Kapitel 4 erörtert. Wie oben bereits erwähnt, besteht die im Sinne minimaler Fehlerwahrscheinlichkeit optimale Strategie zur Entscheidung zwischen verschiedenen Klassen darin, sich für die Klasse mit maximaler a posteriori Wahrscheinlichkeit zu entscheiden. Gute Klassifikatoren werden also diese Strategie möglichst genau zu approximieren versuchen.

Muster im Kontext

Das obige Beispiel der Klassifikation von Schriftzeichen legt sofort die Idee nahe, ein Muster falls möglich und sinnvoll unter Berücksichtigung des **Kontextes** anderer Muster zu klassifizieren und damit die Zuverlässigkeit der gesamten Klassifikation zu verbessern. Dieses ist in Bild 1.5.2 angedeutet. Beispiele dafür sind insbesondere zusammenhängend geschriebene Schrift (wie in Bild 1.5.3) oder zusammenhängend gesprochene Sprache. Es wird also nicht *eine* Klasse Ω_κ für *ein* Muster gesucht, sondern eine *Folge* von Klassen $\boldsymbol{\Omega}$ für eine *Folge* von Mustern. Je Muster wird wieder ein Vergleich zwischen Prototyp und neuem Muster durchgeführt. Zusätzlich ist nun aber die Zuordnung des Merkmalsvektors ${}^{\varrho}\boldsymbol{c}$ zur Klasse Ω_κ unter Berücksichtigung der anderen Zuordnungen zu optimieren. Da es für N beobachtete Muster k^N verschiedene Möglichkeiten gibt, sie einer von k Klassen zuzuordnen, liegt ein komplexes Problem vor, das durch effiziente Suchalgorithmen gelöst werden muss. Die Prinzipien dafür werden in Abschnitt 1.6 beschrieben und in Abschnitt 1.6.8 und Abschnitt 1.6.9 näher ausgeführt.

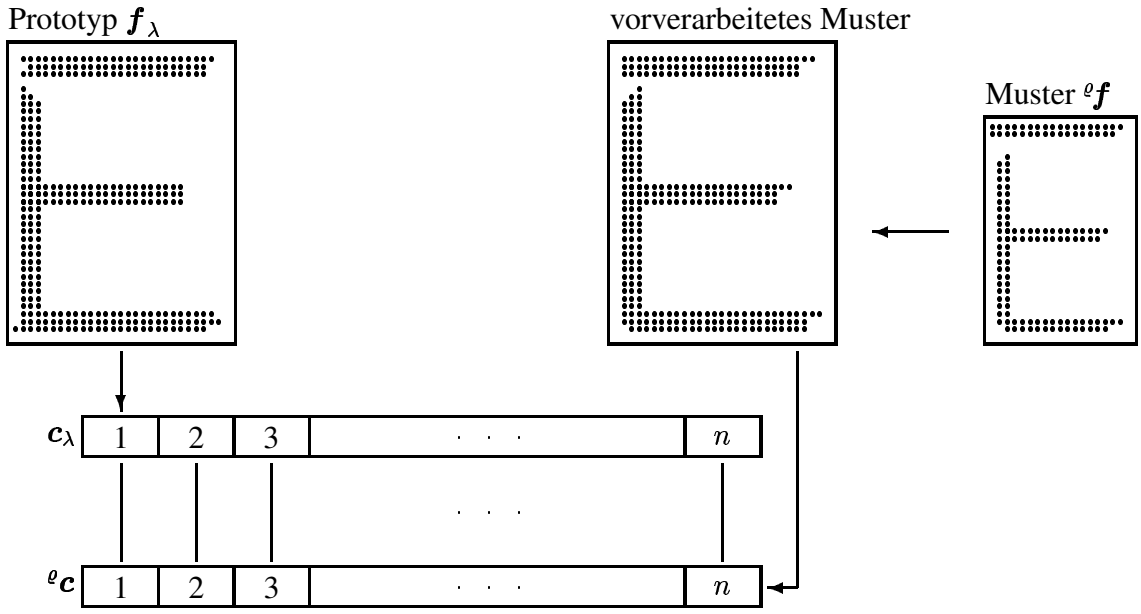


Bild 1.5.1: Klassifikation eines einzelnen Merkmalsvektors

Textur

Texturen sind charakteristische Oberflächeneigenschaften, die von Bildpunkt zu Bildpunkt betrachtet inhomogen sind, in größerem Zusammenhang gesehen aber einen einheitlichen Eindruck vermitteln, wenn auch keinen konstanten Grauwert oder keine konstante Farbe. Beispiele für Texturen zeigt Bild 1.5.4. Sie können im Prinzip wie „ein Muster als Ganzes“ behandelt werden, d. h. es wird, u. U. aus einem kleinen Bildausschnitt, ein Merkmalsvektor berechnet und klassifiziert.

Gesprochene Wörter

Die Klassifikation gesprochener Wörter in einer Äußerung, die noch weitere Wörter enthalten kann, bringt das zusätzliche Problem, dass i. Allg. die Zeiten für Beginn und Ende eines Wortes unbekannt sind. Dieses ist in Bild 1.5.5 gezeigt. Auch hier werden wieder die a posteriori Wahrscheinlichkeiten der alternativen Entscheidungen berechnet. Das erfordert eine stochastische Modellierung, die zeitliche Zusammenhänge, nichtlineare zeitliche Verzerrungen sowie unbekannte Anfangs– und Endpunkte erfassen kann und die eine effiziente Berechnung der a posteriori Wahrscheinlichkeiten sowie ein weitgehend automatisches Training der Modelle erlaubt.

Das allgemeine Problem besteht darin, in einer längeren gesprochenen Äußerung alle vorkommenden Wörter zu erkennen, wobei Anfang, Ende und Zahl der Wörter nicht gegeben sind. Dieses Problem der Erkennung zusammenhängend gesprochener Sprache wird ebenfalls mit statistischen Ansätzen gelöst. Ein Beispiel für eine gesprochene Äußerung zeigt Bild 1.5.6.

Dreidimensionale Objekte

Die Klassifikation eines dreidimensionalen Objektes unbekannter Lage unter Verwendung einer zweidimensionalen Ansicht zeigt Bild 1.5.7. Beispiele zeigt Bild 1.5.8. Für die automati-

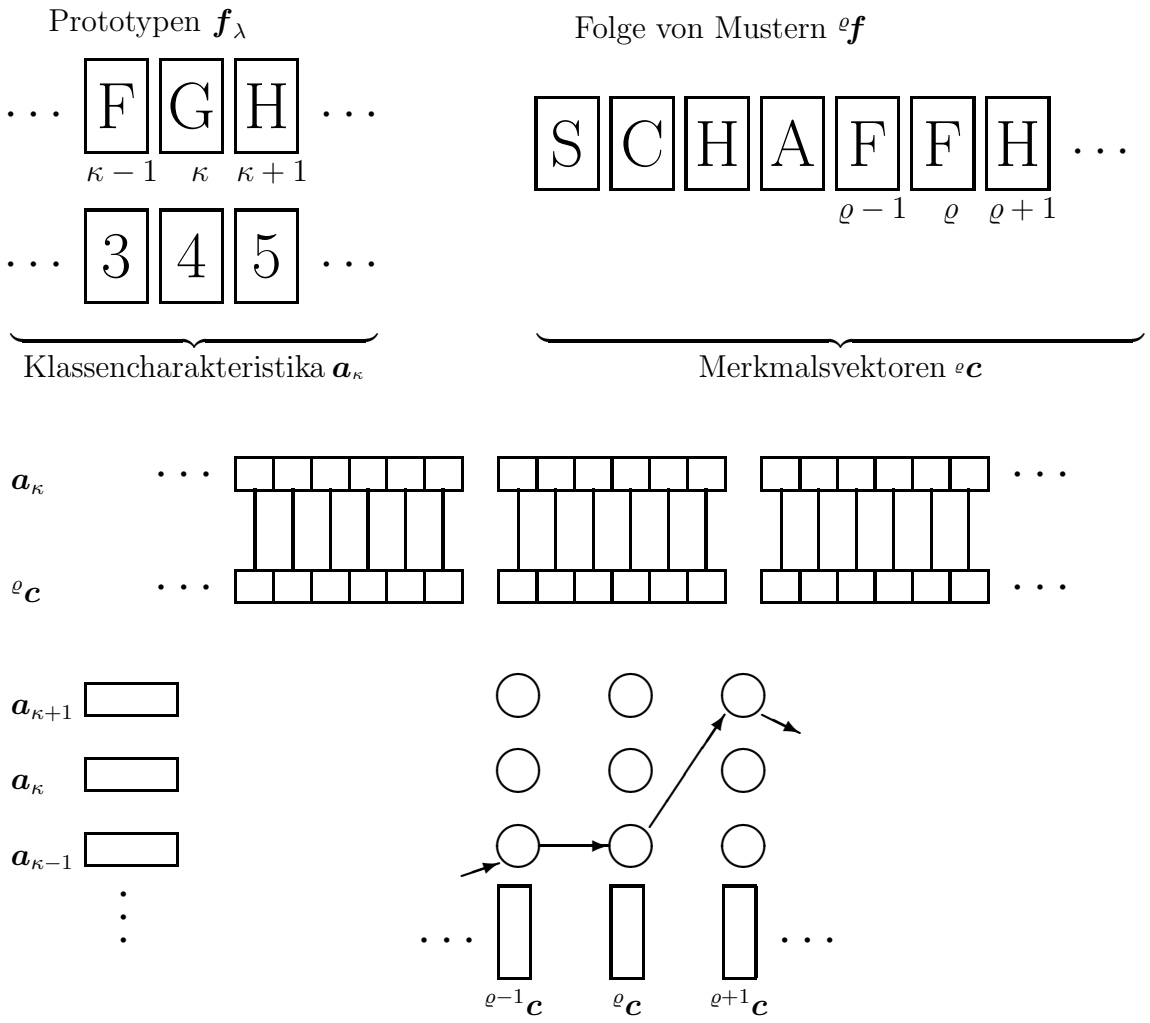


Bild 1.5.2: Klassifikation unter Berücksichtigung des Kontextes anderer Muster

Martensstr. 3
D-91058 Erlangen

Martensstr. 3
D-91058 Erlangen

Bild 1.5.3: Muster (Schriftzeichen) im Kontext

sche Klassifikation sind vor allem die vom Menschen geschaffenen Objekte, weniger bisher die natürlichen Objekte wie Bäume, geeignet. Die unbekannte Lage wird im Raum durch sechs Parameter ${}^{\varrho}\boldsymbol{\theta}_\kappa$ für Translation und Rotation definiert. Im Bild ist dargestellt, dass Algorithmen, die das Klassifikationsproblem durch Zuordnung von einfacheren Bestandteilen wie Linien oder Vertizes lösen, zuvor das Lokalisationsproblem lösen müssen. Zudem entspricht natürlich die Reihenfolge der einfacheren Bestandteile im Modell i. Allg. nicht der Reihenfolge, in der diese im Bild gefunden werden. Es liegt also ein komplexes kombinatorisches Problem vor.

Die Vielfalt dreidimensionaler Objekte ist enorm. Sie reicht von Abmessungen im Bereich von μm bis km; sie umfasst starre und flexibel verformbare Objekte, bewegliche und unbewegliche, Einzelobjekte und Objektgruppierungen, maschinell hergestellte und in der Natur vorkom-

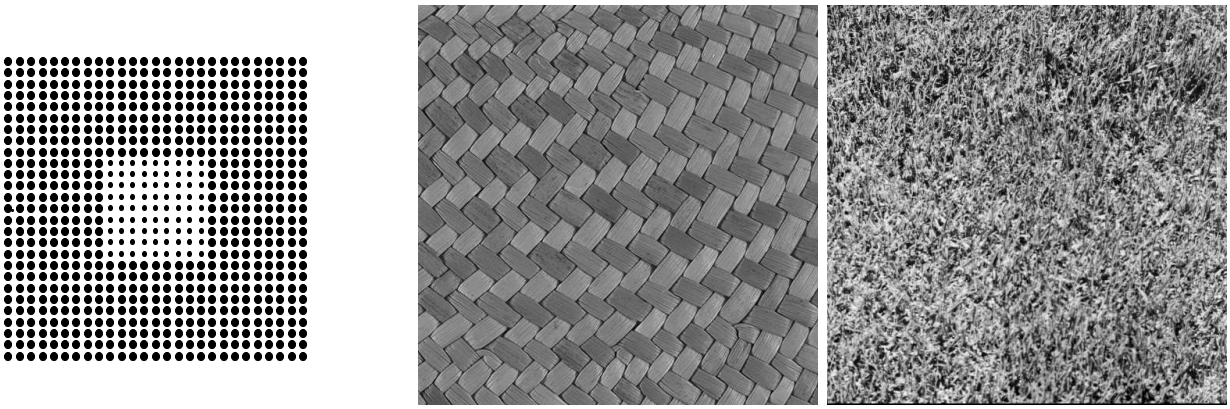


Bild 1.5.4: Eine Textur, die mit zwei Texturelementen konstruiert wurde, sowie zwei natürliche Texturen

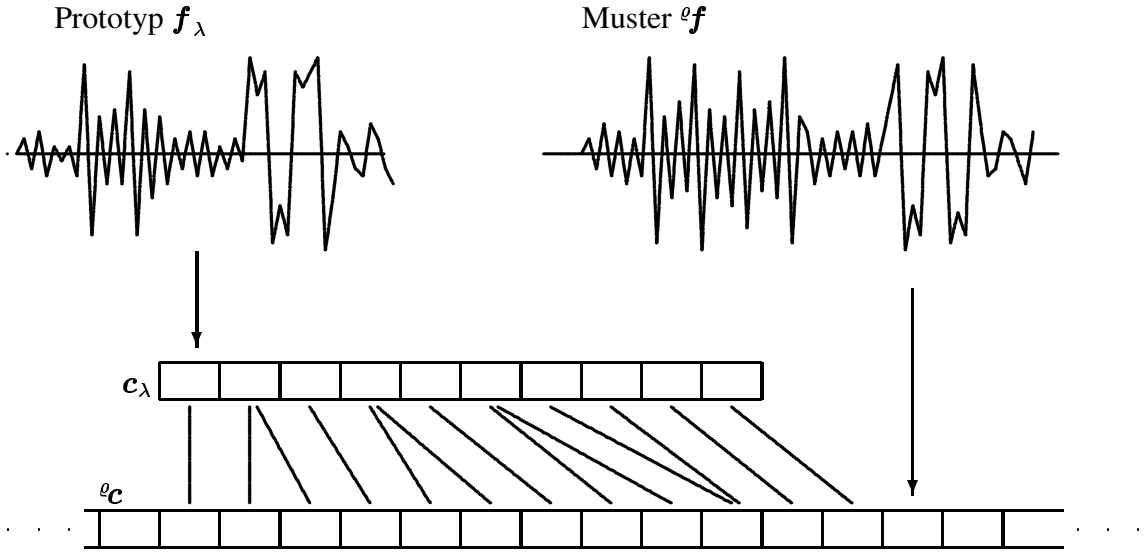


Bild 1.5.5: Die Klassifikation von Wörtern unbekannter Anfangs– und Endzeit

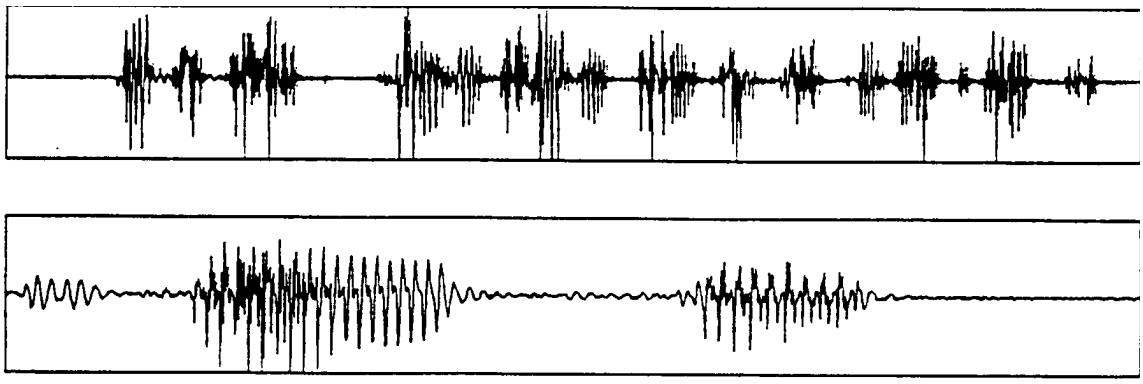


Bild 1.5.6: Ein Sprachsignal; oben die Äußerung „Guten Tag, wann geht morgen vormittag ein Zug nach Frankfurt?“, unten das Wort „Frankfurt“

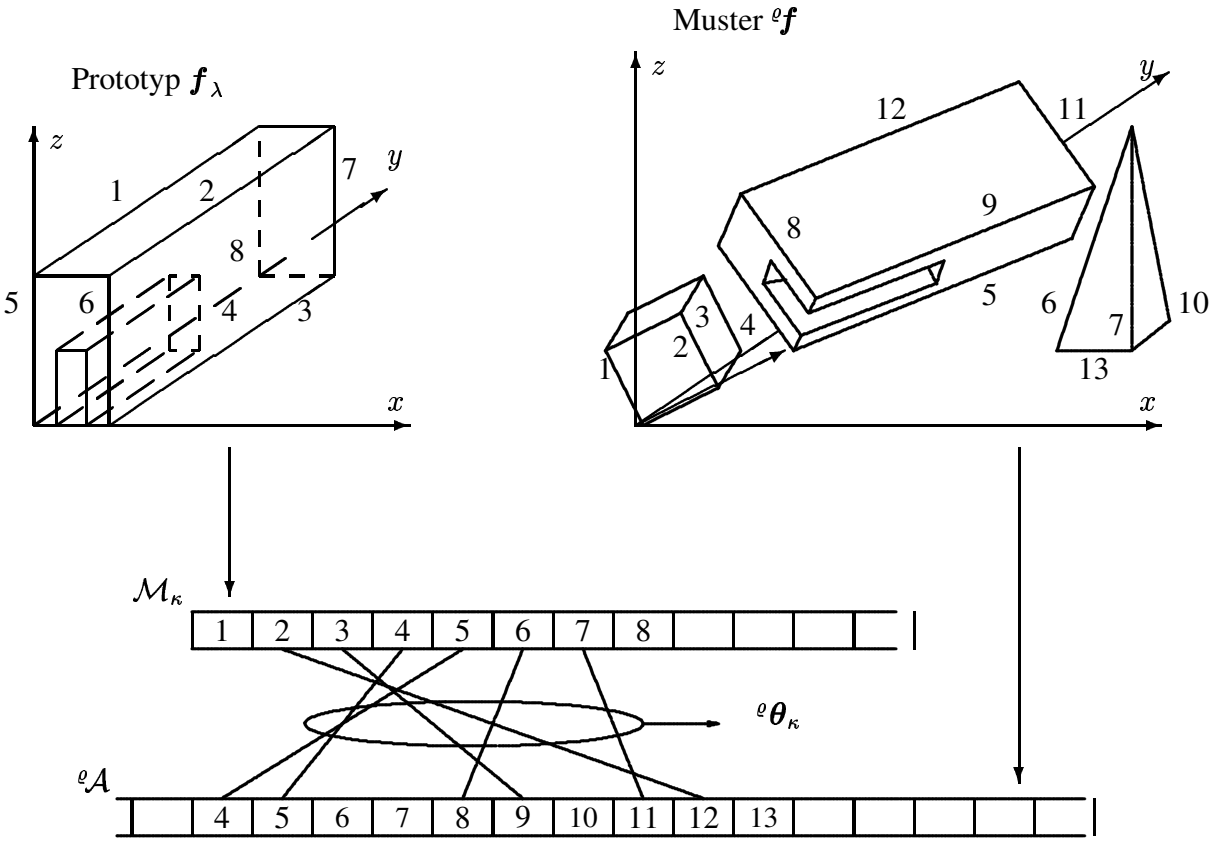


Bild 1.5.7: Klassifikation eines dreidimensionalen Objektes mit i. Allg. unbekannter Lage

Bild 1.5.8: Beispiele für Objekte

mende sowie solche mit matten, spiegelnden und transparenten Oberflächen unter unterschiedlichen Beleuchtungsverhältnissen. Es gibt daher (bisher) nicht *das* Objekterkennungssystem, das für alle diversen Objekte geeignet ist.

Muster in Aufnahmen verschiedener Brennweite

Die Klassifikation eines i. Allg. dreidimensionalen Objektes unter Auswertung zweier Bildfolgen unterschiedlicher Auflösung zeigt Bild 1.5.9. Wir stellen dieses als ein weiteres Problem vor, werden es jedoch nicht im Rahmen dieses Buches behandeln, da auf der statistischen Ent-

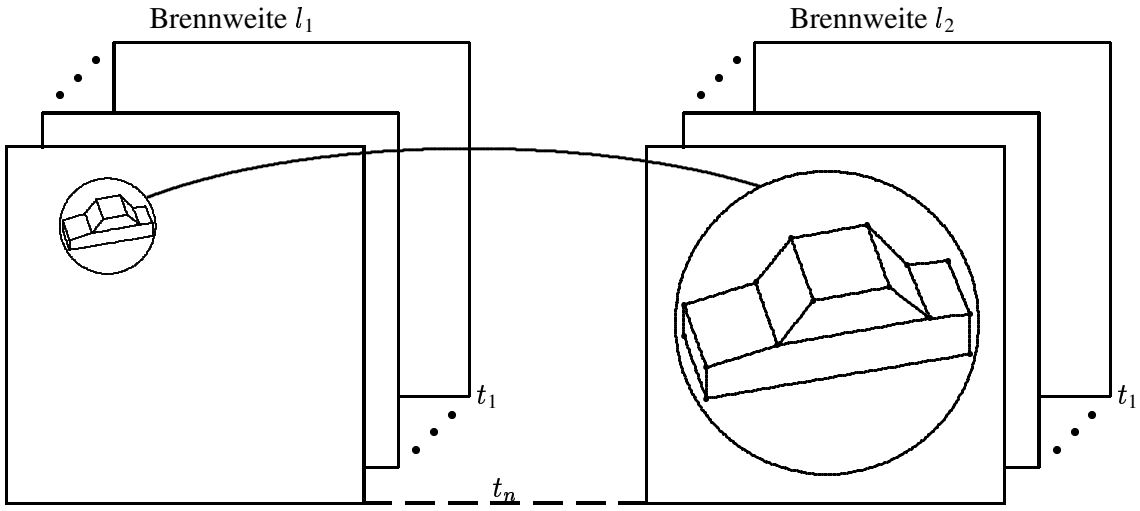


Bild 1.5.9: Klassifikation mit zwei Bildfolgen unterschiedlicher Auflösung

scheidungstheorie basierende Lösungen noch ausstehen. Die Verallgemeinerung ist die Klassifikation eines Objekts, das in verschiedenen Aufnahmen mit verschiedenen Sensoren erfasst wurde. Einen möglichen Ansatz dafür bietet die im Zusammenhang mit (1.3.9) skizzierte Vorgehensweise.

1.6 Optimierungsverfahren

Wenn die freien Bewegungen mit den notwendigen Bedingungen nicht bestehen können, werden sie von der Natur so modifiziert, wie der rechnende Mathematiker nach der Methode der „kleinsten Quadrate“ Erfahrungen ausgleicht. (GAUSS)

(Es) lässt sich der Schluss ziehen, dass Gene, die eine optimale Funktionseinheit repräsentieren, nicht per Zufall entstehen konnten, sondern das Ergebnis eines auf ein Optimum ausgerichteten zielstrebigen Prozesses sein müssen. (EIGEN)

Ein allgemeines Prinzip in Naturwissenschaft und Technik ist die Rückführung von Phänomenen und Anforderungen auf Optimierungsprobleme (s. z. B. (1.3.7), (1.3.8) und (1.3.9)) und die Ableitung von Lösungen durch Optimierungsverfahren. Zum Beispiel

- führt die Berechnung des zeitoptimalen Lichtweges in Medien unterschiedlicher Lichtgeschwindigkeit auf das Brechungsgesetz,
- stellt die Klassifikation von Mustern mit minimalen Kosten ein Optimierungsproblem, wie in Abschnitt 4.1 gezeigt wird,
- ist nach EIGEN die biologische Evolution ein Optimierungsprozess.

Zu einem Optimierungsproblem gehört die Definition der *Menge möglicher Lösungen* und des *Optimierungskriteriums*, das minimiert (oder maximiert) werden soll. Die Menge möglicher Lösungen kann explizit gegeben sein, z. B. ein endliches Intervall der reellen Achse bei der Wahl der Quantisierungskennlinie in Abschnitt 2.1.3, oder implizit durch Angabe von Operationen, mit denen insbesondere aus bekannten Startlösungen neue generiert werden können. Das Optimierungskriterium ist im einfachsten Falle eine ein– oder zweimal differenzierbare Funktion mit *einem* Minimum, z. B. der mittlere quadratische Fehler der Quantisierung in Abschnitt 2.1.3, oder auch eine nichtdifferenzierbare Funktion mit *mehreren* Minima. Für alle

Probleme stehen Lösungsansätze unterschiedlicher Komplexität zur Verfügung. Im Folgenden werden einige wichtige Optimierungsverfahren kurz vorgestellt, wobei natürlich auf Einzelheiten verzichtet wird, da dieses kein Buch über Optimierungsverfahren ist.

1.6.1 Lokale Optimierung ohne Nebenbedingungen

Der einfachste und praktisch auch sehr wichtige Fall ist die Bestimmung des Minimums einer Funktion $g(\boldsymbol{x})$ mit stetigen ersten und zweiten Ableitungen

$$g^* = \min_{\boldsymbol{x}} g(\boldsymbol{x}) \;, \quad \text{oder} \tag{1.6.1}$$

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x}} g(\boldsymbol{x}) \;. \tag{1.6.2}$$

Die Untersuchung von Minimierungsproblemen reicht, da die Maximierung auf eine Minimierung zurückgeführt werden kann durch

$$g^* = \max_{\boldsymbol{x}} g(\boldsymbol{x}) = -\min_{\boldsymbol{x}} -g(\boldsymbol{x}) \;, \quad \text{oder} \tag{1.6.3}$$

$$\boldsymbol{x}^* = \operatorname*{argmax}_{\boldsymbol{x}} g(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{x}} -g(\boldsymbol{x}) \;. \tag{1.6.4}$$

Ein Minimum kann *lokal* oder *global* sein. Das Finden globaler Minima ist wesentlich schwieriger als das lokaler.

Satz 1.1 *Hinreichende Bedingungen für ein lokales Minimum sind*

$$\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^*) = \begin{pmatrix} \frac{\partial g}{\partial x_1} \\ \dots \\ \frac{\partial g}{\partial x_n} \end{pmatrix} = \mathbf{0} \;, \tag{1.6.5}$$

$$\boldsymbol{y}^\mathsf{T} \nabla^2_{\boldsymbol{x}} g(\boldsymbol{x}^*) \boldsymbol{y} = \boldsymbol{y}^\mathsf{T} \begin{pmatrix} \frac{\partial^2 g}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 g}{\partial x_1 \partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial^2 g}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 g}{\partial x_n \partial x_n} \end{pmatrix} \boldsymbol{y} > 0 \;, \quad \forall \boldsymbol{y} \neq \mathbf{0} \;. \tag{1.6.6}$$

Beweis*: s. z. B. [Fletcher, 1987], Chap. 2.*

Die Matrix $\nabla^2 g(\boldsymbol{x}^*)$ in (1.6.6) ist die HESSE-*Matrix*. Die Bedingungen (1.6.5), (1.6.6) bedeuten, dass im Punkt $\boldsymbol{x}^*$ die Funktion g die Steigung Null und nicht negative Krümmung in jeder Richtung haben muss. Die Bedingung (1.6.6) besagt auch, dass die HESSE-Matrix für $\boldsymbol{x} = \boldsymbol{x}^*$ *positiv definit* sein muss; dies ist z. B. der Fall, wenn *alle* ihre Eigenwerte positiv sind.

Eine wichtige Klasse von Gütefunktionen sind die, die den mittleren quadratischen Fehler zwischen einer geschätzten und einer idealen Größe bewerten. Bei quadratischen Gütefunktionen bietet zudem das **Orthogonalitätsprinzip** (vgl. S. 167) eine elegante Alternative.

Beispiele für diese Verfahren sind die Wahl der Quantisierungskennlinie bei der Puls Code Modulation, Satz 2.5, S. 71, der Entwicklungskoeffizienten einer Reihenentwicklung, Satz 3.1, S. 167, oder der Parametermatrix eines verteilungsfreien Klassifikators, Satz 4.16, S. 372.

1.6.2 Iterative Optimierung

Eine Basis vieler iterativer Optimierungsverfahren ist durch den Fixpunktsatz von BANACH gegeben. Er besagt, dass unter einigen Bedingungen ein kontraktiver Operator g, bzw. vereinfacht eine geeignete Funktion g, einen eindeutigen *Fixpunkt* $\boldsymbol{x}^*$ besitzt. Er wird als Grenzwert der Folge $\boldsymbol{x}_0 = \boldsymbol{x},\ \boldsymbol{x}_{i+1} = g(\boldsymbol{x}_i),\ i = 1, 2, \ldots$ bestimmt.

Gradienten- und Koordinatenabstieg

Wenn das lokale Extremum einer Funktion $h(x)$ gesucht ist, so führt die Anwendung der Beziehung $x = g(x)$ auf die Gleichungen

$$\begin{aligned} 0 &= \frac{\partial h(x)}{\partial x} = \beta \frac{\partial h(x)}{\partial x}\ , \\ x &= x + \beta \frac{\partial h(x)}{\partial x} = g(x)\ , \\ x_{i+1} &= x_i + \beta \frac{\partial h(x)}{\partial x} \\ \boldsymbol{x}_{i+1} &= \boldsymbol{x}_i + \beta_i \boldsymbol{r}_i\ . \end{aligned} \tag{1.6.7}$$

Die iterative Bestimmung eines lokalen Extremwertes ist also möglich, indem man mit einer geeigneten *Schrittweite* β_i in eine geeignete *Richtung* $\boldsymbol{r}_i$ geht. Die Bestimmung der Schrittweite wird in der Regel experimentell vorgenommen. Als Richtung wird beim **Gradientenabstieg** wie im obigen Beispiel der Gradient der zu minimierenden Funktion verwendet. Eine Alternative ist der **Koordinatenabstieg**, bei dem der Extremwert für eine Koordinate bestimmt wird, dann für die nächste usw. Eine Anwendung des Gradientenabstiegs ist z. B. der Fehlerrückführungsalgorithmus zum Training neuronaler Netze in Satz 4.17, S. 389.

1.6.3 Lokale Optimierung mit Nebenbedingungen

Die allgemeine Form der Optimierung mit Nebenbedingungen ist

$$g^* = \min_{\boldsymbol{x}} g(\boldsymbol{x})\ , \quad \text{mit } \boldsymbol{x} \in S\ , \tag{1.6.8}$$

d. h. es kommen nur Lösungen aus einer gegebenen Menge S in Frage; diese Einschränkung entfiel in (1.6.1).

Definition 1.11 *Ist $g(\boldsymbol{x})$ eine* konvexe Funktion *und S eine* konvexe Menge*, so liegt ein Problem der* **konvexen Optimierung** *vor.*

Die Funktion $g(\boldsymbol{x})$ ist konvex, wenn

$$g(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda g(\boldsymbol{x}) + (1-\lambda) g(\boldsymbol{y})\ , \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n\ . \tag{1.6.9}$$

Die Menge $S \subseteq \mathbb{R}^n$ ist konvex, wenn

$$\boldsymbol{x}, \boldsymbol{y} \in S\ ,\ \lambda, \mu \geq 0\ ,\ \lambda + \mu = 1\ \Rightarrow\ \lambda \boldsymbol{x} + \mu \boldsymbol{y} \in S\ . \tag{1.6.10}$$

Die Menge S ist ein konvexer Kegel, wenn

$$\boldsymbol{x}, \boldsymbol{y} \in S\ ,\ \lambda, \mu \geq 0\ \Rightarrow\ \lambda \boldsymbol{x} + \mu \boldsymbol{y} \in S\ . \tag{1.6.11}$$

Ein Standardfall ist die Optimierung einer Funktion mit **Nebenbedingungen** n_i in Form von *Gleichungen*

$$g^* = \min_{\boldsymbol{x}} g(\boldsymbol{x}) \,, \tag{1.6.12}$$

$$n_i(\boldsymbol{x}) = 0 \,, \quad i \in I \,. \tag{1.6.13}$$

Mit LAGRANGE-**Multiplikatoren** $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_N)^\mathsf{T}$ wird die LAGRANGE-**Gleichung**

$$\boxed{L(\boldsymbol{x}, \boldsymbol{\vartheta}) = g(\boldsymbol{x}) + \sum_{\nu \in I} \vartheta_\nu n_\nu(\boldsymbol{x})} \tag{1.6.14}$$

gebildet.

Satz 1.2 *Wenn $\boldsymbol{x}^*$ ein Minimum von g unter den Nebenbedingungen n_i ist, gibt es* LAGRANGE-*Multiplikatoren ϑ_ν^*, sodass für diese $\boldsymbol{x}^*$, ϑ_ν^* gilt*

$$\mathbf{0} = \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\vartheta}) = \frac{\partial L(\boldsymbol{x}, \boldsymbol{\vartheta})}{\partial \boldsymbol{x}} \,, \tag{1.6.15}$$

$$0 = \nabla_{\boldsymbol{\vartheta}} L(\boldsymbol{x}, \boldsymbol{\vartheta}) \,, \tag{1.6.16}$$

Für konvexes L sind die obigen Bedingungen hinreichend für ein Minimum.
Beweis*: s. z. B. [Fletcher, 1987], Chap. 9.*

Da im Minimum die Nebenbedingungen verschwinden, ist dort $L(\boldsymbol{x}^*, \boldsymbol{\vartheta}^*) = g(\boldsymbol{x}^*)$.

Eine Verallgemeinerung dieses Optimierungsproblems ist die Verwendung von Nebenbedingungen n_i in Form von Gleichungen *und* Nebenbedingungen n_j in Form von *Ungleichungen*

$$g^* = \min_{\boldsymbol{x}} g(\boldsymbol{x}) \,, \tag{1.6.17}$$

$$n_i(\boldsymbol{x}) = 0 \,, \quad i \in I \,, \tag{1.6.18}$$

$$n_j(\boldsymbol{x}) \geq 0 \,, \quad j \in J \,. \tag{1.6.19}$$

Je nach Typ von g und n_i, n_j ergeben sich unterschiedliche Probleme mit unterschiedlichen Lösungsverfahren. Das speziellste Problem, nämlich lineare Programmierung, liegt vor, wenn alle Funktionen *linear* sind. Das allgemeinste, nämlich semi–definite Programmierung, liegt vor, wenn mit positiv–semidefiniten Matrizen $\boldsymbol{A}_i$, $\boldsymbol{B}$, $\boldsymbol{X}$ gilt, $g = \mathrm{Sp}(\boldsymbol{BX})$, $n_i = \mathrm{Sp}(\boldsymbol{A}_i\boldsymbol{X}) - b_i$, $n_j = \boldsymbol{X}$, wobei mit $\boldsymbol{X} \geq 0$ eine positiv–semidefinite Matrix bezeichnet wird. Dazwischen liegt die quadratische Programmierung, bei der g und n_i quadratische Funktionen sind. Beispiele für konvexe Funktionen sind die stückweise linearen, die quadratischen ($g = \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{q}^\mathsf{T}\boldsymbol{x} + c$ mit $\boldsymbol{Q} \geq 0$) und die stückweise quadratischen Funktionen. Auch die Funktionen $g = \mathrm{Sp}(\boldsymbol{X})$, $\boldsymbol{X} = \boldsymbol{X}^\mathsf{T}$ und $g = -\log|\boldsymbol{X}|$, $\boldsymbol{X} = \boldsymbol{X}^\mathsf{T}$, $\boldsymbol{X} \geq 0$ sind konvex, wobei $|\boldsymbol{X}|$ die Determinante der Matrix bezeichnet. Für alle diese Klassen von Optimierungsproblemen steht Software zu ihrer Lösung zur Verfügung.

Konvexe Optimierungsprobleme der in (1.6.17), (1.6.18) angegebenen Art lassen sich durch die Einführung von LAGRANGE-**Multiplikatoren** $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_N)^\mathsf{T}$ lösen. Damit wird die LAGRANGE-**Gleichung**

$$L(\boldsymbol{x}, \boldsymbol{\vartheta}) = g(\boldsymbol{x}) - \sum_{\nu \in I,J} \vartheta_\nu n_\nu(\boldsymbol{x}) \tag{1.6.20}$$

gebildet. Die Lösung beruht (unter zusätzlichen Regularitätsannahmen) auf folgendem Satz.

Satz 1.3 *(KARUSH–KUHN–TUCKER-Bedingungen) Wenn $\boldsymbol{x}^*$ ein lokales Minimum von (1.6.17) unter den angegebenen Nebenbedingungen ist, dann gibt es LAGRANGE-Multiplikatoren ϑ_ν^*, sodass $\boldsymbol{x}^*, \vartheta_\nu^*$ den folgenden Bedingungen genügen*

$$\begin{aligned}
\mathbf{0} &= \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\vartheta}) \,, && (1.6.21)\\
0 &= n_i(\boldsymbol{x}) \,, \quad i \in I \,, && (1.6.22)\\
0 &\leq n_j(\boldsymbol{x}) \,, \quad j \in J \,, && (1.6.23)\\
0 &\leq \vartheta_i \,, \quad i \in I \,, && (1.6.24)\\
0 &= \vartheta_\nu n_\nu(\boldsymbol{x}) \,, \quad \forall \nu \in I, J \,. && (1.6.25)
\end{aligned}$$

Beweis*: s. z. B. [Fletcher, 1987], Chap. 9.*

Die KARUSH–KUHN–TUCKER-Bedingungen sind *notwendig* für ein Minimum. Wenn zudem die HESSE-Matrix der LAGRANGE-Gleichung (1.6.20) positiv definit ist, dann ist $\boldsymbol{x}^*$ ein lokales Minimum. Aus (1.6.25) geht hervor, dass im Minimum *nicht gleichzeitig* ϑ_ν *und* $n_\nu(\boldsymbol{x})$ von Null verschieden sein können. Man bezeichnet Nebenbedingungen mit $n_\nu = 0$ als **aktive Nebenbedingungen**. Ist in einer aktiven Nebenbedingung $\vartheta_\nu > 0$, ist sie *stark aktiv*, für $\vartheta_\nu = 0$ *schwach aktiv*.

Optimierungsprobleme mit Nebenbedingungen treten z. B. bei den Support Vektor Maschine in Abschnitt 4.3 auf.

1.6.4 EM–Algorithmus

Der *"expectation–maximization"–Algorithmus*, oder kurz **EM–Algorithmus**, ist ein iterativer Algorithmus zum Schätzen statistischer Parameter $\boldsymbol{B}$ einer Verteilungsdichte $p(\boldsymbol{x}|\boldsymbol{B})$ unter Verwendung von *beobachtbaren* Daten $\boldsymbol{x}$ und *verborgenen* bzw. *nicht* beobachtbaren Daten $\boldsymbol{y}$. Er bestimmt ein lokales Minimum. Der Ansatz

„*beobachtete Daten = vollständige Daten − verborgene Daten*"

wird durch die Gleichungen

$$\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{B}) &= \frac{p(\boldsymbol{x},\boldsymbol{B})}{p(\boldsymbol{B})}\,\frac{p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{B})}{p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{B})} = \frac{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{B})}{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{B})}\\
L(\boldsymbol{x},\boldsymbol{B}) &= \log p(\boldsymbol{x}|\boldsymbol{B}) = \log p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{B}) - \log p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{B}) \qquad (1.6.26)
\end{aligned}$$

formalisiert. Das Iterationsverfahren wird mit einem Startwert $\widehat{\boldsymbol{B}}^{(0)}$ begonnen. Im Schritt $i+1$ wird $L(\boldsymbol{x}|\widehat{\boldsymbol{B}}^{(i+1)})$ als Zufallsvariable aufgefasst, von der der bedingte Erwartungswert über die verborgenen Daten $\boldsymbol{y}$ unter Berücksichtigung der beobachteten Daten $\boldsymbol{x}$ und des Schätzwertes $\widehat{\boldsymbol{B}}^{(i)}$ berechnet wird zu

$$\begin{aligned}
E\left\{L\Big(\boldsymbol{x},\widehat{\boldsymbol{B}}^{(i+1)} \mid \boldsymbol{x},\widehat{\boldsymbol{B}}^{(i)}\Big)\right\} &= \int p\Big(\boldsymbol{y}|\boldsymbol{x},\widehat{\boldsymbol{B}}^{(i)}\Big)\,\log p\Big(\boldsymbol{x},\boldsymbol{y}|\widehat{\boldsymbol{B}}^{(i+1)}\Big)\,\mathrm{d}\boldsymbol{y}\\
&\quad - \int p\Big(\boldsymbol{y}|\boldsymbol{x},\widehat{\boldsymbol{B}}^{(i)}\Big)\,\log p\Big(\boldsymbol{y}|\boldsymbol{x},\widehat{\boldsymbol{B}}^{(i+1)}\Big)\,\mathrm{d}\boldsymbol{y}\\
&= Q\Big(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)}\Big) - H\Big(\widehat{\boldsymbol{B}}^{(i+1)}|\widehat{\boldsymbol{B}}^{(i)}\Big)\,. \qquad (1.6.27)
\end{aligned}$$

initialisiere Parameter mit $\widehat{\boldsymbol{B}}^{(0)}$, setze $i = -1$
setze $i = i + 1$
E–Schritt: berechne $Q\left(\widehat{\boldsymbol{B}}^{(i+1)}\middle\vert\widehat{\boldsymbol{B}}^{(i)}\right)$
M–Schritt: berechne $\widehat{\boldsymbol{B}}^{(i+1)} = \underset{\{\widehat{\boldsymbol{B}}^{(i+1)}\}}{\operatorname{argmax}} Q\left(\widehat{\boldsymbol{B}}^{(i+1)}\middle\vert\widehat{\boldsymbol{B}}^{(i)}\right)$
UNTIL $\widehat{\boldsymbol{B}}^{(i+1)} = \widehat{\boldsymbol{B}}^{(i)}$
geschätzter Parameter: $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{B}}^{(i)}$

Bild 1.6.1: Der EM–Algorithmus iteriert die Schritte “expectation” (E–Schritt) und “maximization” (M–Schritt) bis zur Kovergenz gegen ein lokales Optimum

Der Term $Q(\cdot)$ wird als Q–Funktion bzw. als KULLBACK–LEIBLER-**Statistik** bezeichnet.

Satz 1.4 *Zur Maximierung von L, d. h. zur* Maximum-likelihood-Schätzung *von $\boldsymbol{B}$ ist die Maximierung der* KULLBACK-LEIBLER-*Statistik Q hinreichend.*
Beweis*: s. z. B. [Dempster et al., 1977], [Hornegger, 1996], Kapitel 3.*

Wenn statt einer Beobachtung $\boldsymbol{x}$ eine Stichprobe $\{{}^1\boldsymbol{x}, \ldots, {}^N\boldsymbol{x}\}$ von Beobachtungen gemacht wurde, erhält man die KULLBACK–LEIBLER-Statistik Q der Stichprobe aus den KULLBACK–LEIBLER-Statistiken ${}^\varrho Q$ der einzelnen Beobachtungen gemäß

$$Q\left(\widehat{\boldsymbol{B}}^{(i+1)}\middle\vert\widehat{\boldsymbol{B}}^{(i)}\right) = \sum_{\varrho=1}^{N} {}^\varrho Q\left(\widehat{\boldsymbol{B}}^{(i+1)}\middle\vert\widehat{\boldsymbol{B}}^{(i)}\right) . \tag{1.6.28}$$

Daraus ergibt sich das Grundschema des EM–Algorithmus in Bild 1.6.1. Zur Erarbeitung einer konkreten Version müssen die Daten $\boldsymbol{x}, \boldsymbol{y}$ und die Dichten $p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{B}), p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{B})$ ermittelt werden; dann muss die KULLBACK–LEIBLER-Statistik Q berechnet und ein geeignetes Maximierungsverfahren dafür bestimmt werden. Gegenüber einer direkten Maximum-likelihood-Schätzung erscheint der EM–Algorithmus daher zunächst wie ein Umweg. Der Vorteil des EM–Algorithmus liegt jedoch darin, dass das hochdimensionale Maximum-likelihood-Schätzproblem oft in Teilprobleme separiert wird, wodurch die Komplexität reduziert wird, und dass es für die Nullstellen des Gradienten von Q oft geschlossene Lösungen gibt.

Ein wichtiges Beispiel für die Nutzung des EM–Algorithmus ist die Schätzung der Parameter einer Mischungsverteilung in Abschnitt 4.8.2.

1.6.5 Stochastische Approximation

Die Verfahren der **stochastischen Approximation** eignen sich für Minimierungsprobleme des Typs

$$g(\boldsymbol{a}^*) = \min_{\boldsymbol{a}}\{g(\boldsymbol{a})\} , \qquad g(\boldsymbol{a}) = E\{s(\boldsymbol{a}, \boldsymbol{x})\} . \tag{1.6.29}$$

Es wird also der *Erwartungswert* einer Funktion s minimiert, die von der Variablen $\boldsymbol{x}$ und dem Parameter $\boldsymbol{a}$ abhängt. Dieses ist zum einen möglich, wenn man vollständige statistische

Information in Form der Verteilungsdichten vorliegen hat, oder aber wenn man *Beobachtungen* der Variablen $\boldsymbol{x}$ vornehmen kann. Im letzteren Falle kann man den optimalen Parametervektor mit der stochastischen Approximation sogar *ohne* Berechnung des Erwartungswertes iterativ bestimmen. Ausgehend von einem beliebigen Startwert $\boldsymbol{a}_0$ wird bei Beobachtung des ϱ-ten Wertes ${}^{\varrho}\boldsymbol{x}$ der Variablen $\boldsymbol{x}$ ein verbesserter Wert

$$\boldsymbol{a}_{\varrho+1} = \boldsymbol{a}_{\varrho} - \beta_{\varrho} \nabla_{\boldsymbol{a}} s(\boldsymbol{a}_{\varrho}, {}^{\varrho}\boldsymbol{x}) \tag{1.6.30}$$

berechnet. Man beachte, dass in (1.6.30) der Erwartungswert aus (1.6.29) *nicht* mehr auftritt. Im Prinzip wird er durch die Beobachtungen ersetzt, wie es ja auch in der bekannten Schätzgleichung für den Mittelwert einer Zufallsvariablen der Fall ist, nämlich

$$\begin{aligned} m &= E\{x\} = \int_{-\infty}^{\infty} x\, p(x)\, \mathrm{d}x \;, \\ \widehat{m} &= \frac{1}{N} \sum_{\varrho=1}^{N} {}^{\varrho}x \;\approx\; m \;. \end{aligned} \tag{1.6.31}$$

Die in (1.6.30) auftretende Folge β_{ϱ} muss den Bedingungen

$$\beta_{\varrho} \geq 0 \;, \quad \sum_{\varrho=1}^{\infty} \beta_{\varrho} = \infty \;, \quad \sum_{\varrho=1}^{\infty} \beta_{\varrho}^2 < \infty \tag{1.6.32}$$

genügen. Mit den weiteren Bedingungen, dass sich ∇s in der Nähe der Lösung wie eine lineare Funktion verhält und beschränkte Streuung hat, konvergiert die obige Iteration.

Satz 1.5 *Die Iteration gemäß (1.6.30) konvergiert unter den genannten Bedingungen mit Wahrscheinlichkeit 1 und im quadratischen Mittel.*
Beweis*: s. z. B. [Gladyshev, 1965].*

Die stochastische Approximation wird in Abschnitt 4.4.3 als eine Möglichkeit zur Bestimmung der Parametermatrix eines Polynomklassifikators erwähnt, jedoch wird kein weiterer Gebrauch davon gemacht. Ihre Konvergenzgeschwindigkeit ist ein bekanntes Problem, dem mit besonderen Maßnahmen begegnet werden muss.

1.6.6 Globale Optimierung

In den obigen Verfahren wurde (stillschweigend) vorausgesetzt, dass die zu untersuchende Funktion *ein* Minimum hat; dann fallen lokales und globales Minimum zusammen. Wenn eine Funktion *mehrere* Maxima und Minima hat, liefern die obigen Verfahren eines oder mehrere der lokalen Minima, z. B. wenn (1.6.5) mehrere Nullstellen liefert. Ein Ansatz zur Bestimmung des globalen Minimums ist ein **passives Überdeckungsverfahren**, das im Prinzip ein Gitter über den Suchraum legt. In jedem Gitterbereich wird nach einem lokalen Minimum gesucht, unter den lokalen dann das minimale als globales Minimum ausgewählt. Diese Verfahren können kombiniert werden mit einer schrittweisen Verfeinerung der *Auflösung* und/oder einer anfänglichen Suche in einem geeigneten Unterraum mit geringerer Anzahl von Dimensionen, um den Rechenaufwand zu reduzieren.

gegeben: Zustandsmenge $S = \{s_0, s_1, s_2, \ldots, s_i, \ldots\}$, Kostenfunktion $\phi(s_i)$		
bestimme eine *Kontrollsequenz* (T_k) mit den folgenden Eigenschaften: $T_k > 0 \wedge \forall k \geq 0 : T_k \geq T_{k+1} \wedge \lim_{k\to\infty} T_k = 0$		
bestimme den initialen Zustand s_0 und berechne $\phi_0 = \phi(s_0)$		
setze $s_c = s_0$; $\phi_c = \phi_0$		
$\forall$ T_k		
	berechne $s_n =$ erzeuge_zustand(s_c) und $\phi_n = \phi(s_n)$	
	$a =$ akzeptiere_zustand$(\phi_c, \phi_n, T_k) \in \{\mathrm{T}, \mathrm{F}\}$	
	IF	$a = \mathrm{T}$ /* *d.h. wenn neuer Zustand akzeptiert wird* */
	THEN	setze $s_c = s_n$, $\phi_c = \phi_n$
		/* neuen Zustand und Kosten übernehmen */
Ergebnis: Zustand s_c mit den Kosten ϕ_c ist Lösung		

Bild 1.6.2: Prinzip eines Algorithmus zur kombinatorischen Optimierung. Beispiele für die Berechnung von a gibt (1.6.33).

1.6.7 Kombinatorische Optimierung

Wir fassen unter dem Begriff **kombinatorische Optimierung** eine Klasse von Verfahren zusammen, die ausgehend von einem Startwert neue Funktionswerte nach einem Zufallsmechanismus bestimmen bis ein „akzeptables“ Minimum gefunden wurde. Sie setzen weder eine differenzierbare Kostenfunktion $\boldsymbol{g}$ noch eine mit nur einem Minimum voraus und werden auch als *Selektionsverfahren* bezeichnet. Es gibt Probleme, wie z. B. das „Problem des Handlungsreisenden“, die NP–schwer sind, d. h. exponentielle Komplexität haben. In solchen Fällen braucht man aber oft nicht die optimale Lösung, sondern nur eine recht gute suboptimale, die dann mit weit geringerer Komplexität berechenbar ist. Auch dafür eignen sich kombinatorische Optimierungsverfahren.

Ein kombinatorisches Optimierungsproblem wird definiert durch eine (endliche oder unendliche) Menge S von *Zuständen*, eine Kostenfunktion ϕ, die jedem Zustand aus S nichtnegative Kosten zuweist, einen Startzustand $s_0 \in S$, eine Vorschrift zur Generierung neuer Zustände aus gegebenen Zuständen und ein Abbruchkriterium für die Optimierung. Die Zustände sind im Prinzip beliebig, es können symbolische Datenstrukturen, reelle Zahlen oder eine endliche Menge ganzer Zahlen sein. Die Kostenfunktion muss vom Anwender für den jeweiligen Problemkreis gewählt werden. Die Generierung neuer Zustände beruht z. B. auf der zufälligen Generierung kleiner Abweichungen vom aktuell gegebenen Zustand. Das Abbruchkriterium beruht auf dem Erreichen eines Zustands mit minimalen Kosten und kann kombiniert werden mit dem Verbrauch einer vorgegebenen Rechenzeit.

Das Grundschema eines kombinatorischen Optimierungsalgorithmus zeigt Bild 1.6.2. Die dort auftretende *Kontrollsequenz* T_k liefert eine reelle Zahl, deren Wert die Annahme eines neu generierten Zustands steuert. Im Prinzip muss T_k anfänglich hoch sein und dann langsam gegen Null gehen. Damit die Optimierung nicht in einem lokalen Minimum hängen bleibt, werden gelegentlich auch Folgezustände mit höheren Kosten akzeptiert. Prinzipiell bedeutet die monoton mit der Zahl k der Iterationen sinkende Folge der Werte von T_k, dass die Wahrscheinlichkeit für die Annahme eines Folgeszustands mit höheren Kosten mit zunehmender Zahl von Iterationen sinkt. Die Funktionen *erzeuge_zustand* bzw. *akzeptiere_zustand* generieren einen neuen

Zustand bzw. entscheiden, ob dieser als besserer Zustand akzeptiert wird. Bekannte Optimierungsverfahren wenden dafür unterschiedliche Strategien an.

Wir beschränken uns hier auf die Angabe einiger Beispiele für die Annahme neuer Zustände. Eines der bekanntesten Verfahren ist das **simulierte Ausfrieren** ("simulated annealing"), das dem Abkühlen der Schmelze eines Festkörpers nachempfunden ist. Ein Zustand minimaler Kosten entspricht dabei einem Erstarren in einem regelmäßigen Kristallgitter. Die mit der Zeit t_k fallende Temperatur $T(t_k)$ der Schmelze liefert die Kontrollsequenz T_k. Bei geeigneter Vorgehensweise, insbesondere geeignetem Abkühlen, lässt sich das Erreichen eines globalen Minimums beweisen. Die Annahme eines neuen Zustands geht aus (1.6.33) hervor. Danach wird ein neuer Zustand s_n immer dann angenommen, d. h. $a = \mathrm{T}$, wenn seine Kosten ϕ_n kleiner sind als die Kosten ϕ_c des aktuellen Zustands s_c. Hat der Folgezustand größere Kosten, wird er mit der Wahrscheinlichkeit $\exp[-\Delta\phi/T_k]$ angenommen. Das wird dadurch erreicht, dass q in (1.6.33) eine im Intervall $[0, 1]$ gleichverteilte Zufallsvariable ist. Ein Vorschlag für die Wahl der Kontrollsequenz ist $T_k = T_0\alpha^k\,,\ \alpha \approx 0,95$. Bei der **Schwellwertakzeptanz** ("threshold acceptance") werden alle Folgezustände akzeptiert, deren Kosten nicht wesentlich über denen des aktuellen Zustands liegen. Die Kontrollsequenz bestimmt den Schwellwert für die zulässige Kostendifferenz. Im Unterschied zur Schwellwertakzeptanz, die alle Folgezustände mit nicht zu hoher Kosten*differenz* akzeptiert, werden beim **Sintflutalgorithmus** ("great deluge algorithm") alle Folgezustände mit nicht zu hohen Kosten akzeptiert. Schließlich wird bei der **stochastischen Relaxation** gar keine Kontrollsequenz verwendet. Es werden nur Folgezustände mit geringeren Kosten akzeptiert, wodurch die Gefahr des Hängenbleibens in einem lokalen Minimum besteht. Kriterien für die Annahme eines Folgeszustandes sind damit

$$a = \begin{cases} \mathrm{T} : \begin{cases} \Delta\phi = \phi_n - \phi_c \le 0 \quad \text{oder} \\ \Delta\phi > 0 \ \wedge\ q \le \exp[-\Delta\phi/T_k] \end{cases} & \text{simuliertes Ausfrieren}\,, \\ \mathrm{T} : (\phi_n - \phi_c) \le T_k & \text{Schwellwertakzeptanz}\,, \\ \mathrm{T} : \phi_n \le T_k & \text{Sintflut}\,, \\ \mathrm{T} : \phi_n \le \phi_c & \text{stochastische Relaxation}\,, \\ \mathrm{F} : \text{sonst}\,. \end{cases} \tag{1.6.33}$$

Die kombinatorischen Optimierungsverfahren sind äußerst leistungsfähig, vorausgesetzt sie werden sorgfältig an das Problem angepasst. Dieses erfordert genaue Kenntnisse der theoretischen Grundlagen der Optimierungsverfahren und der Besonderheiten der Anwendung.

1.6.8 Dynamische Programmierung

Die **dynamische Programmierung** (DP) ist ein Optimierungsverfahren, bei dem eine Folge von *Entscheidungen* getroffen wird. Jede Entscheidung transformiert den aktuellen *Zustand* in einen neuen. Es ist eine Folge von Entscheidungen gesucht, die eine *Kostenfunktion minimiert* (oder eine Gütefunktion maximiert). Wesentlich für die DP ist die Annahme, dass das **Optimalitätsprinzip** gilt, welches besagt:

> In einer Folge optimaler Entscheidungen ist jede Teilfolge selbst eine optimale Folge von Entscheidungen.

Die Folge von Entscheidungen lässt sich als *Pfad* in einem geeignet definierten Netzwerk von Zuständen auffassen. Das Optimalitätsprinzip bedeutet dann, dass jeder Teilpfad des optimalen

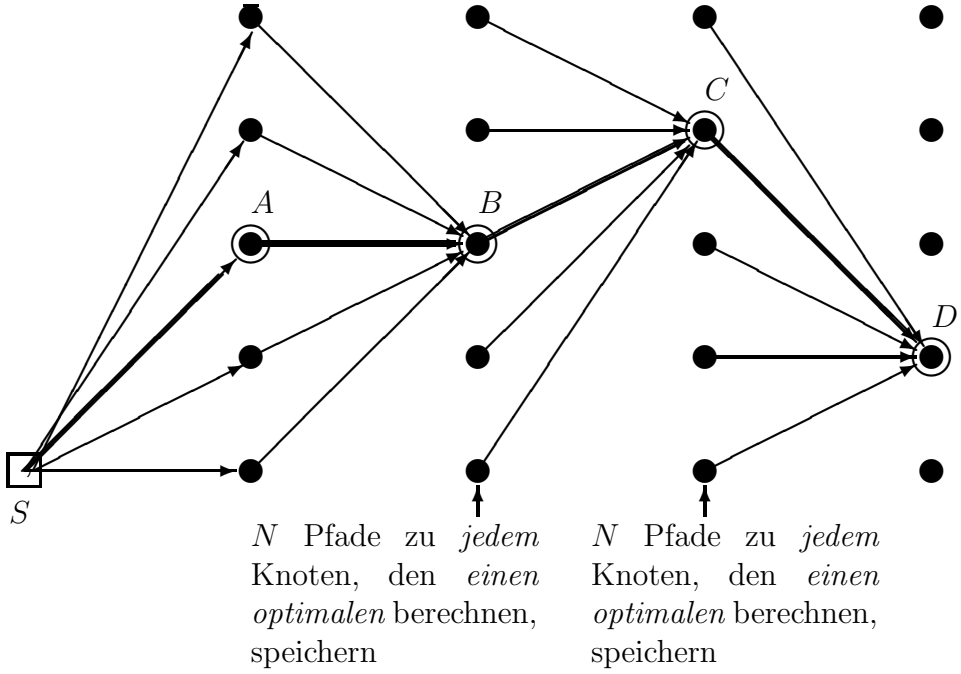


Bild 1.6.3: Der optimale Pfad von S nach D führe über A, B, C. Wenn man den optimalen Pfad von S nach B berechnet hat, kann mal alle nichtoptimalen Pfade nach B löschen, da sie wegen des Optimalitätsprinzips auch später nicht mehr zum optimalen Pfad von S nach D über B gehören können.

Pfades selbst ein optimaler Pfad ist. Die Folge davon ist, dass die *Komplexität* des Entscheidungsprozesses drastisch reduziert wird, und zwar von exponentiell auf linear in der Zahl der Suchschritte. Bild 1.6.3 verdeutlicht das an einem Zustandsnetzwerk. Wenn das Optimalitätsprinzip *nicht* gilt und man den optimalen Pfad aktuell von S über A nach B bestimmt hat, kann bei einer Fortsetzung der Suche über B hinaus z. B. der optimale Pfad von S nach C *nicht mehr* über A gehen. Daher muss man *alle* Pfade von S nach B auch für spätere Suchschritte aufheben und bei der Suche berücksichtigen – die Suchkomplexität ist exponentiell mit der Zahl der Suchschritte. Wenn dagegen das Optimalitätsprinzip gilt, ist die Suchkomplexität nur linear mit der Zahl der Suchschritte, da man nach Erreichen von B *alle* Pfade bis auf den einen optimalen von S nach B nicht mehr weiter zu betrachten braucht. Die Gültigkeit des Optimalitätsprinzips ergibt sich aus folgendem Satz.

Satz 1.6 *Das Optimalitätsprinzip gilt für* monotone *und* separierbare *Kostenfunktionen.*

Die Monotonie sichert, dass bei einer Verlängerung des Suchpfades die Kosten nicht abnehmen können. Die Separierbarkeit hat zur Folge, dass sich die Kosten zerlegen lassen in die bisher angefallenen und die im nächsten Suchschritt entstehenden. Ein typisches Beispiel einer monotonen und separierbaren Kostenfunktion ist die Summe nichtnegativer Kostenanteile.

In Anwendungen in der Mustererkennung haben wir oft eine endliche Menge $S = \{S_1, S_2, \ldots, S_I\}$ von Zuständen, die in mehreren Rechenschritten durchsucht werden, um einen Pfad mit minimalen Kosten vom Startzustand in den Endzustand zu finden, wie Bild 1.6.4 zeigt. Bezeichnet man die Kosten zur Erreichung des Zustands S_j im Rechenschritt n mit $\varphi(n, j)$ und

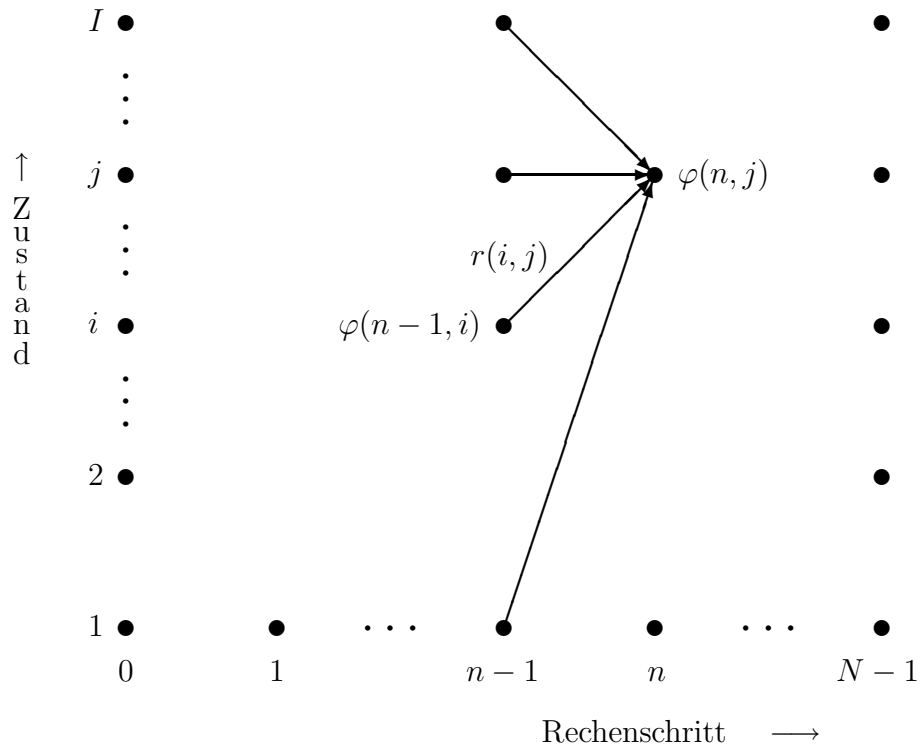


Bild 1.6.4: Im Rechenschritt n werden rekursiv die minimalen Kosten berechnet, um vom Startzustand in den Zustand S_j zu gelangen

gegeben: Menge S der Zustände, Kosten $r(i,j)$ beim Übergang von S_i nach S_j, Anfangszustand S_1, Endzustand S_I
gesucht: Pfad minimaler Kosten vom Anfangs– zum Endzustand
initialisiere: Kosten im Anfangszustand $\varphi(0,1)=0$, $\varphi(0,i)=\infty$, $i=2,\ldots,I$
FOR $n=1,\ldots,N-1$ DO:
FOR $i=1,\ldots,I$ DO:
$\varphi(n,j)=\min_i\{r(i,j)+\varphi(n-1,i)\}$
$\varphi_{\min}=\varphi(N-1,I)$

Bild 1.6.5: Prinzip des Algorithmus für die Berechnung minimaler Kosten mit der dynamischen Programmierung

die Kosten beim Übergang von S_i nach S_j mit $r(i,j)$, so ist die wesentliche Rekursionsgleichung zur Berechnung der Kosten

$$\varphi(n,j)=\min_i\{r(i,j)+\varphi(n-1,i)\}\ . \tag{1.6.34}$$

Wegen des Optimalitätsprinzips hängt sie nur von den Kosten φ zur Erreichung der Zustände im Schritt $n-1$ und den Übergangskosten r ab, aber *nicht* auch noch von Kosten zur Erreichung von Zuständen in den Rechenschritten $n-2, n-3, \ldots$. Daraus resultiert ein einfacher Algorithmus, der in einer Doppelschleife über alle Zustände und über alle Rechenschritte zur Berechnung der Kosten geht, wie Bild 1.6.5 zeigt.

Die DP ist in vielen Anwendungen ein unverzichtbares Werkzeug geworden, in der Mustererkennung z. B. in der Kontextberücksichtigung (s. Abschnitt 4.7)

1.6.9 Graph– und Baumsuche

Grundsätzlich lassen sich die Schritte bei der Lösung eines Problems durch Knoten und Kanten in einem Graphen oder Baum darstellen. Der Startknoten ist das anfänglich gegebene Problem, hier also die Klassifikation oder Analyse eines gegebenen Musters. Die Anwendung irgendeiner, hoffentlich zur Problemlösung beitragenden, Operation ergibt einen neuen Knoten, der mit seinem Vorgänger durch eine Kante verbunden ist. Der Zielknoten enthält eine *optimale Lösung* des Problems, wobei Optimalität wieder im Sinne eines problemangepassten Optimierungskriteriums verstanden wird, hier also Klassifikation oder Analyse mit minimalem Fehler und u. U. minimaler Rechenzeit. Man sucht nach einer *Folge von Operationen*, die mit minimalen Kosten zu einer optimalen Lösung führt; diese Folge ist ein Pfad im Graphen, nämlich der optimale Lösungspfad. Prinzipiell hat man also die Güte der Lösung (z. B. die Fehlerrate eines zu entwickelnden Systems zur Klassifikation von Mustern) von den Kosten des Lösungspfades (z. B. der Zeit zur Entwicklung des optimalen Systems zur Klassifikation von Mustern) zu unterscheiden.

Von den verschiedenen Algorithmen zur Graphsuche deuten wir hier nur den A^*–*Algorithmus* an. Bei der Suche wird ein Such*baum* generiert, der implizit definiert ist, d. h. es ist der leere Startknoten v_0 gegeben sowie Operationen, mit denen zu einem gegebenen Knoten weitere generiert werden können. Es ist *nicht* das Ziel, *alle* Knoten des Baumes zu generieren, sondern möglichst nur die, die zum Finden des optimalen Lösungspfades erforderlich sind. Wir betrachten nun irgendeinen Knoten v_i im Suchbaum und den *optimalen Pfad* vom Startknoten v_0 über diesen Knoten v_i zum Zielknoten v_g. Für die Kosten dieses Pfades werden drei Bedingungen gestellt. Die Kosten $\varphi(v_i)$ des optimalen Pfades über v_i müssen sich als erste Bedingung *additiv* zusammensetzen aus den Kosten $\psi(v_i)$ von v_0 nach v_i und den Kosten $\chi(v_i)$ von v_i nach v_g

$$\varphi(v_i) = \psi(v_i) + \chi(v_i) \; . \tag{1.6.35}$$

In einem konkreten Suchproblem werden diese unbekannten Kosten durch eine *Schätzung* $\widehat{\varphi}(v_i) = \widehat{\psi}(v_i) + \widehat{\chi}(v_i)$ ersetzt. Es wird gefordert, dass die Schätzung der Restkosten $\widehat{\chi}(v_i)$ *optimistisch* ist in dem Sinne, dass der Schätzwert *kleiner* als die tatsächlichen Kosten ist

$$\widehat{\chi}(v_i) \leq \chi(v_i) \; . \tag{1.6.36}$$

Wenn $r(v_i, v_k)$ die tatsächlichen Kosten eines optimalen Pfades von v_i zu einem Nachfolger v_k sind, muss als drittes die *Monotoniebedingung*

$$\widehat{\chi}(v_i) - \widehat{\chi}(v_k) \leq r(v_i, v_k) \tag{1.6.37}$$

gelten.

Ein Graphsuchalgorithmus mit einer Kostenfunktion, die den obigen drei Bedingungen genügt, heisst A*–Algorithmus. Sein Arbeitsprinzip ist informell in Bild 1.6.6 angegeben. Die Analogie zur dynamischen Programmierung (DP) ist offensichtlich. Dort wird ein optimaler Pfad in einem Netzwerk von Zuständen, d. h. in einem Graphen gesucht. Bei der DP muss die Kostenfunktion ebenfalls einschränkenden Bedingungen genügen.

gegeben: Startknoten v_0, Kostenfunktion $\varphi(v_i)$, Operationen zur Generierung von Nachfolgern eines Knotens		
generiere Nachfolger des Startknotens, berechne ihre Kosten und markiere sie als unbearbeitete Knoten		
	IF	ein unbearbeiteter Knoten mit den geringsten Kosten ist ein Zielknoten
	THEN	ENDE mit Erfolg: optimaler Lösungspfad gefunden
	ELSE	expandiere den unbearbeiteten Knoten mit geringsten Kosten, d.h. generiere alle seine Nachfolger
	prüfe, ob ein besserer Pfad zu einem Knoten gefunden wurde	
UNTIL keine unbearbeiteten Knoten mehr vorhanden		
ENDE mit Misserfolg: kein Lösungspfad gefunden		

Bild 1.6.6: Prinzip des A*–Algorithmus zur Suche nach einem optimalen Pfad in einem Graphen

Satz 1.7 *Der A*–Algorithmus* terminiert *immer für endliche Graphen; er terminiert für unendliche, wenn ein Lösungspfad existiert. Er ist* zulässig*, d. h. wenn es überhaupt einen Lösungspfad gibt, terminiert er mit dem optimalen Pfad.*
Beweis*: s. z. B. [Hart et al., 1968, Nilsson, 1982]*

1.6.10 Evolutionäre Algorithmen

Unter dem Begriff *evolutionäre Algorithmen* werden Optimierungsverfahren zusammengefasst, die mit unterschiedlichen Ansätzen Prinzipien der biologischen Evolution nachzubilden versuchen. Dazu gehören *genetische Algorithmen, genetische Programmierung, Evolutionsstrategien* und *evolutionäre Programmierung*; ein weiterer Ansatz sind die *Partikelschwärme* (particle swarm optimization, PSO). Damit lassen sich Optimierungsprobleme behandeln, bei denen die Kostenfunktion nicht differenzierbar sein muss und bei denen auch die Menge der Lösungen nicht explizit gegeben ist. Ein Beispiel für so ein Problem ist die Suche nach der Form eines Tragwerks, das bei vorgegebenem Gewicht und Länge an seinem Ende eine möglichst große Last tragen kann (Kranausleger). Die Angabe einer expliziten Funktion $\boldsymbol{g}(\boldsymbol{x})$ für die *Form* des Tragwerks scheidet hier aus.

Jede Lösung des Optimierungsproblems (jedes Tragwerk) wird als *Individuum* bezeichnet, eine Menge von Lösungen als *Population*. Die Eignung eines Individuums wird durch eine *Kostenfunktion* bewertet. Die Eigenschaften des Individuums charakterisieren die Lösung. Man beginnt mit einer Anfangspopulation, die z. B. zufällig erzeugt wird. Die aktuelle Population erzeugt *Nachkommen* durch zufällige *Kreuzung* ihrer Eigenschaften und durch zufällige *Mutation* (Veränderung) ihrer Eigenschaften. Die Nachkommen werden mit der Kostenfunktion bewertet, und es erfolgt eine *Selektion* der geeignetsten, die die nächste Population bilden. Dieser Prozess der Bildung neuer Populationen wird fortgesetzt bis ein Minimum der Kostenfunktion erreicht ist. Ein evolutionärer Algorithmus kann einige vom Anwender zu wählende Parameter enthalten, wie z. B. die Mutationswahrscheinlichkeit. Es ist prinzipiell möglich, auch diese Parameter in den evolutionären Optimierungsprozess einzubeziehen.

Evolutionäre Algorithmen lassen sich so modifizieren, dass sehr viele Optimierungsproble-

me damit behandelt werden können. Es ist eine andere Frage, ob das auch immer effizient ist. Ihre Stärke liegt wie oben angedeutet darin, dass man damit Optimierungsprobleme lösen kann, die man mit anderen Verfahren *nicht* lösen kann. Im Zusammenhang mit der Mustererkennung kann man evolutionäre Algorithmen zur automatisierten Optimierung von ganzen Systemkonfigurationen einsetzen.

Die *genetischen Algorithmen* arbeiten auf einer *Kodierung* des Problems durch eine Folge von Zeichen (bits). *Genetische Programmierung* nutzt genetische Algorithmen zur Generierung von Programmen die möglichst gut vorgegebene Kriterien erfüllen. Bei den *Evolutionsstrategien* wird das Problem nicht kodiert, sondern direkt auf den Parametern des Anwendungsbereichs gearbeitet. Kreuzung und Mutation werden also auf diese angewendet, nicht auf kodierende „Gene“. Sie sind für den praktischen Einsatz im Bereich der Technik besonders gut zu handhaben. Die *evolutionäre Programmierung* weist damit große Ähnlichkeit auf (und nicht mit der genetischen Programmierung); Unterschiede liegen vor allem in Einzelheiten der Generierung neuer Populationen.

Eine (μ, λ)–Evolutionsstrategie ((μ, λ)–ES) ist das Tupel

$$(\mu, \lambda)-\mathrm{ES} = \left(P_\mu^0,\ \mu,\ \lambda;\ r,\ m,\ s,\ \Delta\sigma,\ \varphi,\ B,\ T\right)\ . \tag{1.6.38}$$

Sie generiert aus einer Anfangspopulation P_μ^0 von μ Eltern λ Nachkommen, von denen wiederum μ selektiert werden, um die nächsten λ Nachkommen zu generieren, usw. In der $(\mu + \lambda)$–Evolutionsstrategie werden die μ Eltern in die Population der Nachfolger aufgenommen. Eine Population im Generationsschritt n besteht aus μ Individuen $a_i^n = (\boldsymbol{d}_i^n,\ \boldsymbol{e}_i^n)\ ,\ i = 1, \ldots, \mu$, die durch Eigenschaften bestimmt sind. Der Vektor von Eigenschaften $\boldsymbol{e}_i^n$ geht in die Kostenfunktion φ ein, während der Vektor $\boldsymbol{d}_i^n$ für jede Komponente ν von $\boldsymbol{e}_i^n$ eine Standardabweichung $\sigma_{i,\nu}^n$ enthält, die ihrerseits durch Mutation verändert wird. Die Standardabweichung $\sigma_{i,\nu}^n$ steuert über eine Normalverteilung die Mutation der Eigenschaften $\boldsymbol{e}_i^n$; die Standardabweichung $\sigma_{i,\nu}^n$ wird ihrerseits mutiert durch eine Normalverteilung mit der Standardabweichung $\Delta\sigma$. Die Kreuzungs–, Mutations– bzw. Selektionsoperatoren sind r, m bzw. s. Zusätzlich können Nebenbedingungen B für die Kostenfunktion definiert werden und muss ein Kriterium T für die Terminierung der Evolution mit einer bestimmten Population angegeben werden. Diese kurze Darstellung zeigt, dass eine Evolutionsstrategie zahlreiche Entwurfsentscheidungen offen lässt, mit denen das Verfahren an das Problem angepasst werden muss und welche die Qualität einer gefundenen Lösung wesentlich beeinflussen.

1.6.11 “No–Free–Lunch”–Theoreme

Man kann daran denken, *das* allgemeine bzw. universelle Optimierungsverfahren zu suchen, mit dem sich *alle* Optimierungsprobleme lösen lassen. Die so genannten “No–Free–Lunch”–Theoreme besagen im Wesentlichen, dass es so ein universelles Optimierungsverfahren *nicht* gibt. Wir verzichten hier auf die Wiedergabe der präzisen und formalen Beschreibung, da uns die kurze Botschaft genügt:

> Man kann, je nach Problemstellung, auf keines der obigen Verfahren verzichten und braucht vielleicht auch noch andere.

1.7 Anwendungen

Die vornehmste aller Illusionen aber ist, dass einem etwas genüge. (JUNG)

Du steckst dir die Grenzen. Es stimmt nicht, dass du dieses nicht tun kannst, sondern du tust es einfach nicht. (KONFUZIUS)

In Kapitel 2 bis Kapitel 4 dieses Buches stehen allgemeine Verfahren zur Klassifikation von Mustern im Vordergrund. Daher wird in diesem Abschnitt eine kurze Darstellung einiger wichtiger Anwendungen von Klassifikationssystemen gegeben. Die Anwendungen werden in sieben Bereiche sowie einige Sonderbereiche unterteilt.

Schriftzeichen: Das automatische Lesen von Schriftzeichen findet z. B. Anwendung bei der Verarbeitung von Rechnungs- und Zahlungsbelegen, der Sortierung von Post und allgemein bei der Verarbeitung von Dokumenten. Auf diesem Gebiet, das zu den „klassischen“ Anwendungen der Mustererkennung gehört, sind seit mehreren Jahren kommerzielle Geräte auf dem Markt. Für Massenanwendungen werden i. Allg. maschinell gedruckte Zeichen, vielfach mit standardisierter Form wie OCR-A und OCR-B, verlangt (s. Bild 1.2.2). Bei solchen Zeichen erreichen moderne Maschinen Lesegeschwindigkeiten und –zuverlässigkeiten, welche die von Menschen weit übertreffen. Es gibt auch Maschinen für das Lesen von handgedruckten Zeichen, jedoch hängt deren Leistung stark von der Art der Auflagen an den Schreiber und dessen Disziplin beim Schreiben ab. Das automatische Lesen zusammenhängend geschriebener Handschrift ist z. B. für das Lesen beliebiger Postsendungen wichtig und wird dort eingesetzt.

Medizinische Versorgung: In der Medizin treten sowohl wellenförmige Muster, wie Elektrokardiogramme (EKG), Phonokardiogramme (PKG) und Elektroenzephalogramme (EEG), als auch bildhafte Muster, wie Röntgen–, Magnetresonanz–, Ultraschall– und Computertomographiebilder, auf. Die auszuwertende Datenmenge ist enorm, die Fehlerhäufigkeit beträchtlich. Eine Entlastung des Personals von Routineaufgaben ist wünschenswert, um mehr Zeit für die kritischen Fälle zu geben. Dazu kommen Aufgaben der Operationsplanung und der rechnergestützten Operationsausführung, die u. a. auch die genaue Vermessung und quantitative Auswertung von Bildern erfordern.

Industrielle Anwendungen: Industrielle Anwendungen ergeben sich im Bereich der Qualitätskontrolle und der Fertigungsautomatisierung. Im ersteren Falle werden Fertigungs– und Montagefehler z. B. aus Laufgeräuschen von Motoren und Getrieben oder durch optische Kontrolle von Schaltungen, Kontakten oder sonstigen Bauteilen ermittelt. Der letztere Fall betrifft die Automatisierung von Fertigungsprozessen mit Hilfe von sensorgesteuerten Robotern, die in begrenztem Umfang ihre Umgebung und Werkstücke „sehen“ und mit ihnen entsprechend agieren können.

Serviceroboter: Im Bereich des Haushaltes, der (häuslichen) Pflege und der Dienstleistungen ergeben sich vielfältige neue Einsatzmöglichkeiten für flexible, lernfähige und mit internem Wissen über den Problemkreis ausgestattete Roboter. Sie sollten ihre Umgebung und relevante Gegenstände erkennen und erwünschte Dienstleistungen erbringen können, Gefahrensituationen vermeiden und durch gesprochene Kommandos steuerbar sein. Es müssen also sowohl sensorische als auch aktorische Fähigkeiten vorhanden sein.

Fahrerassistenzsysteme: Angesichts wachsender Verkehrsdichte einerseits und steigendem Anspruch an den Fahrkomfort andererseits sind Systeme in Entwicklung, die eine aus dem bewegten Fahrzeug aufgenommene Bildfolge des Vekehrsgeschehens interpretieren und daraus sicherheitsrelevante Hinweise für den Fahrer ableiten.

Erdfernerkundung: Mit Aufnahmegeräten an Bord von Flugzeugen oder Satelliten können in kurzer Zeit Daten von großen Teilen der Erdoberfläche gesammelt werden, die außer den

militärischen auch wichtige zivile Anwendungen erlauben. Dazu gehören z. B. geologische Untersuchungen, Land– und Forstwirtschaft (Ernteerträge, Schädlingsbefall), Geographie (Stadtplanung, automatisierte Kartenerstellung), Umweltschutz (Luft– und Wasserverschmutzung), Ozeanographie und Meteorologie. Für diese Zwecke wurden zahlreiche interaktive Systeme entwickelt, in denen versucht wird, Hintergrundwissen und Überblick eines menschlichen Experten mit der Datenverarbeitungs– und –speicherkapazität eines Rechners zu einem möglichst leistungsfähigen Gesamtsystem zu kombinieren.

Spracherkennung: Die Spracherkennung umfasst die Teilgebiete der Klassifikation isoliert gesprochener Wörter, des Erkennens und Verstehens zusammenhängend gesprochener (auch spontaner) Sprache inklusive der Antwort auf gesprochene Fragen bzw. der Übersetzung gesprochener Äußerungen; sie umfasst auch die Identifikation unbekannter Sprecher mit einem gesprochenen Text und die Verifikation (Bestätigung der Identität) von Sprechern. Für die Klassifikation isolierter Wörter werden seit mehreren Jahren kommerzielle Geräte für die Dateneingabe und die Steuerung von Geräten durch einfache Kommandos angeboten und eingesetzt. Kontinuierliche Spracherkennung findet Anwendung in Diktiersystemen sowie in Auskunftssystemen mit gesprochener Sprache für die Ein– und Ausgabe. Die Identifikation und Verifikation von Sprechern hat mögliche Anwendungen in der Kriminalistik und bei der Zugangskontrolle zu Räumen und Gebäuden oder auch der Zugriffskontrolle zu Information.

Sonderbereiche: Spezielle Anwendungen ergeben sich unter anderem im Bereich der Archäologie, der Hochenergiephysik, der Kriminalistik, des Militärs, der Seismologie, der Werkstoffwissenschaften und der Wirtschaftswissenschaften. Ihre Erörterung würde hier jedoch zu weit führen.

Typische Klassifikationsaufgaben sind bei den obigen Anwendungen z. B. die Klassifikation von Schriftzeichen (die Klassen sind hier die Bedeutungen der Zeichen wie „A“ oder „3“), die Auswertung von EKG (die Klassen sind hier die Diagnosen), die Klassifikation von Laufgeräuschen (mögliche Klassen sind hier „einwandfrei“, „fehlerhaft“, „unklar“), die Klassifikation von Bildpunkten in Multispektralaufnahmen (die Klassen sind hier Bodentypen wie „Wald“, „Wiese“ oder „Acker“), die Klassifikation bzw. Erkennung von Gesichtern (die Klassen sind hier die Identitäten von Personen), die Erkennung von Emotionen (mögliche Klassen sind hier z. B. „neutral“, „ärgerlich“ oder „desinteressiert“) und die Klassifikation gesprochener Wörter (die Klassen sind hier die Bedeutungen der Wörter wie „Haus“ oder „fünf“).

1.8 Ausblick

Der Stoff, aus dem unser Wissen besteht, ist per se auch der Stoff aus dem die Welt selbst besteht. Dieser „Stoff“ *ist* Information. (LYRE)

Worüber man nicht reden kann, darüber muss man schweigen. (WITTGENSTEIN)

Ähnlich wie der absolute Nullpunkt in der Tieftemperaturforschung eine asymptotische Größe ist, wird es sich mit den verschiedenen „absoluten Nullpunkten“ in der Mustererkennung verhalten, nämlich

$$\left[\begin{pmatrix} \text{Klassifikationsfehler} \\ \text{Klassifikationszeit} \\ \text{Lokalisationsfehler} \\ \text{Lokalisationszeit} \\ \text{Trainingszeit} \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right], \quad \forall \text{ Problemkreise} . \tag{1.8.1}$$

Ihre Minimierung je Problemkreis liefert Stoff für viele weitere Jahre Forschungs– und Entwicklungsarbeit. Dazu kommen Gesichtspunkte wie der Bedarf an Zeit und Geld für Benutzereinweisung, Wartung und Anpassung an neue Anforderungen.

Es steht die Frage an, wie man als Mensch in einer Vielfalt von Nervensignalen bzw. wie ein mustererkennender Algorithmus mit Merkmalsextraktion, Bildsegmentierung, Sprachsegmentierung usw. zu einem ganzheitlichen Gesamteindruck von einem Bild– oder Sprachsignal kommen kann. Algorithmen und Gleichungen, die dieses leisten, sind z. Z. nicht in Sicht. Allerdings gibt es, ohne Anspruch auf Vollständigkeit, z. B. folgende Mutmaßungen:

1. In der geschlossenen Schleife von Wahrnehmung und Handeln, bzw. von Sensorik und Aktorik ist der ganzheitliche Eindruck ein emergenter Effekt.
2. Es ist der Geist, der sieht, hört und wahrnimmt.
3. Der Quantencomputer wird neue Möglichkeiten eröffnen (vielleicht auch der noch undiskutierte 0-bran-Computer oder noch weitere bisher ungedachte Rechner).
4. Neuronale Netze mit Bewusstsein sind zu entwerfen.
5. Die Aspekte Kreativität, Intuition und Einsicht fehlen heutigen Programmen noch.
6. Das können Maschinen nicht, da sich ein Gehirn nur durch ein Gehirn angemessen simulieren lässt.

Es gibt u. a. die Forschungsthemen Mustererkennung, Künstliche Intelligenz, Bild– und Sprachverstehen, Rechnersehen, aktives Sehen, exploratives Sehen, usw. Bereits vor etwa 1300 Jahren wurde das „wahre Sehen“ (oder die wahre Wahrnehmung) erkannt, ein Thema dessen algorithmische Behandlung, auch aus Sicht der (TURING) Berechenbarkeit, eine vielleicht abschließende Herausforderung darstellt: „Einsicht in das Nichtsein, dies ist das wahre Sehen, das ewige Sehen.“

1.9 Literaturhinweise

Man gebrauche gewöhnliche Worte und sage ungewöhnliche Dinge. (SCHOPENHAUER)

Wer die Literatur nicht liest, muss sie neu erfinden. (ROSENFELD)

In diesem einführenden Kapitel zitieren wir vor allem einige Bücher und Übersichtsartikel in Zeitschriften.

Biologische und physiologische Aspekte der Perzeption werden z. B. behandelt in [Arbib, 1995, Flanagan, 1978, Jameson und Hurvich, 1977, Keidel und Neff, 1974, 1975, 1976, Leuwenberg und Buffart, 1978].

Zur Klassifikation von Merkmalsvektoren sind zahlreiche Bücher erschienen, z. B. [Arkadew und Braverman, 1967, Chen, 1973, Duda und Hart, 1972, Fu und Mendel, 1970, Fu und Whinston, 1977, Fukunaga, 1990, Meyer-Brötz und Schürmann, 1970, Niemann, 1974, Niemann, 1983, Schürmann, 1977, Sebestyen, 1962, Tou und Gonzalez, 1974, Tsypkin, 1973, Young und Calvert, 1973]. Neuronale Netze, unter anderem zur Klassifikation von Mustern, sind das Thema von Büchern wie [Arbib, 1995, Cichocki und Unbehauen, 1994, Ritter et al., 1991, Zell, 1994]. Mit der Problematik der Definition von Klassen befasst sich das Buch [Lakoff, 1987]. Die Erkennung bzw. Klassifikation von Wörtern in gesprochener Sprache ist Gegenstand von Büchern wie [Junqua und Haton, 1996, Rabiner und Juang, 1993, Schukat-Talamazzini, 1995]. Die Erkennung von dreidimensionalen Objekten wird u. a. in [Besl und Jain, 1985, Grimson, 1990, Jain und Flynn, 1993] beschrieben, speziell statistische Ansätze dafür in den Dissertationen [Hornegger, 1996, Pösl, 1999, Reinhold, 2003,

Wells III, 1993]. Die Analyse und wissensbasierte Verarbeitung (die nicht Gegenstand dieses Buches sind) wird zum Beispiel in den Büchern [Hanson und Riseman, 1978, Kazmierczak, 1980, Lea, 1980, Niemann, 1981, Sagerer und Niemann, 1997] behandelt, wobei in [Sagerer und Niemann, 1997] insbesondere auf den in (1.3.8), S. 24, angegebenen Ansatz eingegangen wird. Zum aktiven Sehen, das hier ebenfalls nicht behandelt wird, wird auf [Aloimonos und Weiss, 1987, Bajcsy, 1988, Blake und Yuille, 1992, Paulus, 2001] verwiesen; der in (1.3.9), S. 25, angedeutete Ansatz wird in [Denzler, 2003] an verschiedenen Beispielen konkret durchgeführt.

Zur Optimierung existiert eine umfangreiche Literatur, zu der hier nur ein kleiner Einstieg vermittelt wird. Allgemeine Einführungen sowie die Behandlung iterativer Verfahren geben [Fletcher, 1987, Papageorgiou, 1991]. Konvexe Optimierung wird in [Demyanov und Rubinov, 1995, Hiriart-Urruty und de Lemarechal, 1993, Mangasarian, 1994, Minoux, 1986, Ye, 1997] dargestellt und Anwendungen in der Klassifikation in [Burges, 1998, Schölkopf, 1997]. Die Beschränkung auf binäre Variable wird in [Keuchel et al., 2001] für Bildzerlegung und perzeptuelle Gruppierung betrachtet. Die iterative Schätzung mit dem EM-Algorithmus hat informelle Vorläufer in [Ball und Hall, 1967, Scudder, 1965] und wurde in [Dempster et al., 1977] begründet. Die Darstellung des EM-Algorithmus hier folgt [Hornegger, 1996], eine umfangreiche Darstellung gibt [McLachlan und Krishnan, 1997]. Eine nichtparametrische Version wird in [Zribi und Ghorbel, 2003] vorgestellt. Stochastische Approximation wird in [Albert und Gardner, 1967, Benveniste et al., 1990, Wasan, 1969] behandelt, stochastische Optimierung in [Spall, 2003]. Globale Optimierung wird in [Floudas, 1996, Horst und Tuy, 1990, Toern und Zilinskas, 1989] behandelt. Der Nutzen von mehreren Auflösungsschritten ist in [Heitz et al., 1994] diskutiert. Beispiele für kombinatorische Optimierungsverfahren sind in [Dorigo et al., 1996, Schrimpf et al., 2000] gegeben. Das simulierte Ausfrieren geht auf [Metropolis et al., 1953] zurück und wird in Texten wie [Aarts und Korst, 1989, Laarhoven und Aarts, 1987] behandelt; Beispiele zur Nutzung in der Bildverarbeitung sind [Fischer, 1995, Geman und Geman, 1984, Sontag und Sussmann, 1985]. Für Schwellwertakzeptanz, Sintflut Algorithmus und stochastische Relaxation verweisen wir auf [Dueck und Scheuer, 1990, Dueck, 1993, Geman und Geman, 1984]. Einführungen in dynamische Programmierung geben [Bellman, 1957, Bellmann und Dreyfus, 1962, Dano, 1975, Dreyfus und Law, 1977, Nemhauser, 1969, Schneeweiß, 1974]. Verfahren zur Suche nach optimalen Pfaden in Graphen und Bäumen werden in [Nilsson, 1982, Niemann, 1990, Sagerer und Niemann, 1997] beschrieben. Genetische Algorithmen werden in [Goldberg, 1989, Holland, 1975] beschrieben, genetische Programmierung in [Koza, 1992, Koza, 1994], Evolutionsstrategien in [Rechenberg, 1973, Rechenberg, 1994], evolutionäre Programmierung in [Fogel et al., 1966, Fogel, 1991], Optimierung mit Partikelschwärmen in [Bonabeau et al., 1999, Kennedy und Eberhardt, 1995, Kennedy et al., 2001] und mit (den hier nicht behandelten) Ameisenkolonien in [Dorigo et al., 1996, Dorigo und Stützle, 2004, Dorigo et al., 2004]; eine Übersicht zu evolutionären Algorithmen gibt [Bäck et al., 1997]. Die "No-Free-Lunch"-Theoreme findet man mit Beweis in [Wolpert und Macready, 1997], eine Diskussion der Theoreme in [Culberson, 1996].

Zu den in Bild 1.3.2 verwendeten Bildern:

1. Lena: Dieses Porträt ist ein Ausschnitt aus dem Bild des "playboy's playmate of the month" von Lenna Sjööblom im Playboy, Vol. 19, No. 11, November 1972. Es wird für viele Bereiche der Bildverarbeitung häufig zur Illustration verwendet; das Für und Wider diskutiert z. B. der Leitartikel [Munson, 1996]. Farb- und Grauwertbild sind/waren

verfügbar in http://www.ece.rice.edu/ wakin/images/.

2. COIL-20: Dieses steht für Columbia Object Image Library; es handelt sich um Bilder von 20 (dreidimensionalen) Objekten in 72 äquidistanten Drehlagen, also insgesamt 1440 Bilder, in der Auflösung 128×128. Weitere Beispiele zeigt Bild 3.8.2. Sie ist/war u. a. verfügbar in http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php.
3. MNIST: Es handelt sich um eine Trainingsstichprobe von 60,000 sowie eine Teststichprobe von 10,000 handgeschriebenen Ziffern im Format 28×28, die eine Untermenge aus einer Stichprobe des NIST (National Institute of Standards and Technology) sind; die beiden Stichproben sind disjunkt bezüglich der Schreiber. Sie wurden größennormiert auf 20×20 und dann in einem 28×28 Bild so zentriert, dass der Schwerpunkt im Mittelpunkt liegt. Sie ist/war u. a. verfügbar in http://kernel-machines.org.

Literaturverzeichnis

[Aarts und Korst, 1989] Aarts, E., Korst, J. *Simulated Annealing and Boltzmann Machines. A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley-Interscience Series in Discrete Mathematics and Optimization. J. Wiley, Chichester, 1989.

[Albert und Gardner, 1967] Albert, A.E., Gardner, L.A. *Stochastic Approximation and Nonlinear Regression*. MIT Press, Cambridge, Mass., 1967.

[Aloimonos und Weiss, 1987] Aloimonos, A.B., Weiss, I. Active vision. In *Proc. First Int. Conf. on Computer Vision*, S. 35–54. London, 1987.

[Arbib, 1995] Arbib, M.A., Hg. *The Handbook of Brain-Theory and Neural Networks*. The MIT Press, Cambridge, Massachusetts, USA, 1995.

[Arkadew und Braverman, 1967] Arkadew, A.G., Braverman, E.M. *Teaching Computers to Recognize Patterns*. Academic Press, London, 1967.

[Bäck et al., 1997] Bäck, T., Hammel, U., Schwefel, H.-P. Evolutionary computation: Comments on the history and current state. *IEEE Trans. on Evolutionary Computation*, 1(1):3–17, 1997.

[Bajcsy, 1988] Bajcsy, R. Active perception. *Proc. IEEE*, 76(8):996–1005, 1988.

[Ball und Hall, 1967] Ball, G.H., Hall, J.D. A clustering technique for summarizing multivariate data. *Behavioral Sciences*, 12:153–155, 1967.

[Bellman, 1957] Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, N.J., 1957.

[Bellmann und Dreyfus, 1962] Bellmann, R., Dreyfus, S. *Applied Dynamic Programming*. Princeton Univ. Press, Princeton, 1962.

[Benveniste et al., 1990] Benveniste, A., Métivier, M., Priouret, P. *Adaptive algorithms and stochastic approximations*, Bd. 22 von *Applications of mathematics*. Springer, Berlin, 1990.

[Besl und Jain, 1985] Besl, P.J., Jain, R.C. Three-dimensional object recognition. *ACM Computing Surveys*, 17:75–145, 1985.

[Blake und Yuille, 1992] Blake, A., Yuille, A., Hg. *Active Vision*. The MIT Press, Cambridge, Mass., 1992.

[Bonabeau et al., 1999] Bonabeau, E., Dorigo, M., Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Oxford, England, 1999.

[Burges, 1998] Burges, C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[Chen, 1973] Chen, C.H. *Statistical Pattern Recognition*. Hayden, New York, 1973.

[Cichocki und Unbehauen, 1994] Cichocki, A., Unbehauen, R. *Neural Networks for Optimization and Signal Processing*. J. Wiley, New York, 1994.

[Culberson, 1996] Culberson, J.C. On the futility of blind search. Technical Report TR 96-18, Department of Computing Science, The University of Alberta, Edmonton, Alberta, Canada, 1996.

[Dano, 1975] Dano, S. *Nonlinear and dynamic programming*. Springer, Wien, 1975.

[Dempster et al., 1977] Dempster, A., Laird, N., Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[Demyanov und Rubinov, 1995] Demyanov, V.F., Rubinov, A.M. *Constructive Nonsmooth Analysis*. Verlag Peter Lang, Bern, 1995.

[Denzler, 2003] Denzler, J. *Probabilistische Zustandsschätzung und Aktionsauswahl im Rechnersehen*. Logos Verlag, Berlin, Germany, 2003.

[Dorigo et al., 2004] Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. *Ant Colony Optimization and Swarm Intelligenc, 4th Int. Workshop ANTS 2004 (Brussels, Belgium)*, Bd. LNCS 3172. Springer, Berlin Heidelberg, 2004.

[Dorigo et al., 1996] Dorigo, M., Maniezzo, V., Colorni, A. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26:29–41, 1996.

[Dorigo und Stützle, 2004] Dorigo, M., Stützle, T. *Ant Colony Optimization*. The MIT Press, Cambridge, Mass., 2004.

[Dreyfus und Law, 1977] Dreyfus, S.E., Law, A.M. *The Art and Theory of Dynamic Programming*. Academic Press, New York, 1977.

[Duda und Hart, 1972] Duda, R.O., Hart, P.E. Use of Hough transformation to detect lines and curves in pictures. *Communic. of the Association for Computing Machinery*, 15:11–15, 1972.

[Dueck, 1993] Dueck, G. New optimization heuristics: The great deluge algorithm and the record-to-record-travel. *Journal of Computational Physics*, 104(1):86–92, 1993.

[Dueck und Scheuer, 1990] Dueck, G., Scheuer, T. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics*, 90(1):161–175, 1990.

[Fischer, 1995] Fischer, V. *Parallelverarbeitung in einem semantischen Netzwerk für die wissensbasierte Musteranalyse*, Bd. 95 von *DISKI*. infix, Sankt Augustin, 1995.

[Flanagan, 1978] Flanagan, J.L. *Speech Analysis, Synthesis and Perception*. Springer, New York, 1978.

[Fletcher, 1987] Fletcher, R. *Practical Methods of Optimization*. J. Wiley, Chichester, 2. Aufl., 1987.

[Floudas, 1996] Floudas, C.A., Hg. *State of the art in global optimization*. Kluwer Acad. Publ., Dordrecht, 1996.

[Fogel, 1991] Fogel, D.B. *System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling*. Ginn Press, Needham Heights, 1991.

[Fogel et al., 1966] Fogel, L.J., Owens, A.J., Walsh, M.J. *Artificial Intelligence Through Simulated Evolution*. J. Wiley, New York, 1966.

[Fu und Mendel, 1970] Fu, K.S., Mendel, J.M. *Adaptive, Learning, and Pattern Recognition Systems*. Academic Press, New York, 1970.

[Fu und Whinston, 1977] Fu, K.S., Whinston, A.B., Hg. *Pattern Recognition Theory And Application*. Nordhoff, Leyden, 1977.

[Fukunaga, 1990] Fukunaga, K. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2. Aufl., 1990.

[Geman und Geman, 1984] Geman, S., Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[Gladyshev, 1965] Gladyshev, E.G. On stochastic approximation. *Automatika e Telemekanika*, 10(2):275–278, 1965.

[Goldberg, 1989] Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York, 1989.

[Grimson, 1990] Grimson, W.E.L. *Object Recognition by Computer: The Role of Geometric Constraints*. The MIT Press, Cambridge, MA, 1990.

[Hanson und Riseman, 1978] Hanson, A.R., Riseman, E.M., Hg. *Computer Vision Systems*. Academic Press, New York, 1978.

[Hart et al., 1968] Hart, P.E., Nilsson, N.J., Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[Heitz et al., 1994] Heitz, F., Perez, P., Bouthemy, P. Multiscale minimization of global energy functions in some visual recovery problems. *Computer Vision, Graphics, and Image Processing*, 59:125–

134, 1994.

[Hiriart-Urruty und de Lemarechal, 1993] Hiriart-Urruty, J.-B., de Lemarechal, C. *Convex Analysis and Minimization Algorithms I, II*. Springer, Berlin Heidelberg, 1993.

[Holland, 1975] Holland, J.H. *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press, Ann Arbor, 1975.

[Hornegger, 1996] Hornegger, J. *Statistische Modellierung, Klassifikation und Lokalisation von Objekten*. Berichte aus der Informatik. Shaker Verlag, Aachen, Germany, 1996.

[Horst und Tuy, 1990] Horst, R., Tuy, H. *Global Optimization*. Springer, Berlin, 1990.

[Jain und Flynn, 1993] Jain, A.K., Flynn, P.J., Hg. *Three-Dimensional Object Recognition Systems*. Elsevier, Amsterdam, 1993.

[Jameson und Hurvich, 1977] Jameson, D., Hurvich, L.M. *Visual Psychophysics, Handbook of Sensory Physiology*, Bd. VII/4. Springer, Berlin, Heidelberg, New York, 1977.

[Junqua und Haton, 1996] Junqua, J.-C., Haton, J.-P. *Robustness in Automatic Speech Recognition*. Kluwer Acad. Publ., Boston, 1996.

[Kazmierczak, 1980] Kazmierczak, H. *Erfassung und maschinelle Verarbeitung von Bilddaten*. Springer, Wien, New York, 1980.

[Keidel und Neff, 1974, 1975, 1976] Keidel, W.D., Neff, W.D. *Auditory System, Handbook of Sensory Physiology*, Bd. VI/1-3. Springer, Berlin, Heidelberg, New York, 1974, 1975, 1976.

[Kennedy und Eberhardt, 1995] Kennedy, J., Eberhardt, R.C. Particle swarm optimization. In *Proc. IEEE Int. Conference on Neural Networks*, S. 1942–1948. (IEEE Service Center, Piscataway), Perth, Australia, 1995.

[Kennedy et al., 2001] Kennedy, J., Eberhardt, R.C., Shi, Y. *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, San Francisco, USA, 2001.

[Keuchel et al., 2001] Keuchel, J., Schellewald, C., Cremers, D., Schnörr, C. Convex relaxations for binary image partitioning and percepual grouping. In B. Radig, S. Florczyk, Hg., *Pattern Recognition. Proc. 23rd DAGM Symposium*, S. 353–360. (Springer LNCS 2191, Berling Heidelberg, ISBN 3-540-42596-9), München, Germany, 2001.

[Koza, 1992] Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, MA, 1992.

[Koza, 1994] Koza, J.R. *Genetic Programming II*. The MIT Press, Cambridge, MA, 1994.

[Laarhoven und Aarts, 1987] Laarhoven, P. van, Aarts, E. *Simulated Annealing: Theory and Applications*. D. Reidel Publ. Comp., Dordrecht, 1987.

[Lakoff, 1987] Lakoff, G. *Women, Fire, and Dangerous Things*. University of Chicago Press, Chicago IL, 1987.

[Lea, 1980] Lea, W.A., Hg. *Trends in Speech Recognition*. Prentice Hall, Englewood Cliffs, N. J., 1980.

[Leuwenberg und Buffart, 1978] Leuwenberg, E.L.J., Buffart, H.F.J. *Formal Theories of Visual Perception*. J. Wiley, New York, 1978.

[Mangasarian, 1994] Mangasarian, O.L. *Nonlinear Programming*. SIAM, 1994.

[McLachlan und Krishnan, 1997] McLachlan, G., Krishnan, T. *The EM Algorithm and Extensions*. J. Wiley, New York, 1997.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. Equation of state calculations for fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[Meyer-Brötz und Schürmann, 1970] Meyer-Brötz, G., Schürmann, J. *Methoden der automatischen Zeichenerkennung*. R. Oldenbourg, München, 1970.

[Minoux, 1986] Minoux, M. *Mathematical Programming: Theory and Algorithms*. J. Wiley, Chichester, 1986.

[Munson, 1996] Munson, D.C. A note on lena. *IEEE Trans. on Image Processing*, 5(1):3, 1996.

[Nemhauser, 1969] Nemhauser, G.L. *Einführung in die Praxis der dynamischen Programmierung*. Oldenbourg, München, 1969.

[Niemann, 1974] Niemann, H. *Methoden der Mustererkennung*. Akademische Verlagsgesellschaft, Frankfurt, 1974.

[Niemann, 1981] Niemann, H. *Pattern Analysis*. Springer Series in Information Sciences 4. Springer, Berlin, Heidelberg, New York, 1981.

[Niemann, 1983] Niemann, H. *Klassifikation von Mustern*. Springer; (zweite erweiterte Auflage 2003 im Internet: http://www5.informatik.uni-erlangen.de/MEDIA/nm/klassifikation-von-mustern/m00links.html), Berlin, Heidelberg, 1983.

[Niemann, 1990] Niemann, H. *Pattern Analysis and Understanding*. Springer Series in Information Sciences 4. Springer, Berlin, 2. Aufl., 1990.

[Nilsson, 1982] Nilsson, N.J. *Principles of Artificial Intelligence*. Springer, Berlin, Heidelberg, New York, 1982.

[Papageorgiou, 1991] Papageorgiou, M. *Optimierung*. Oldenbourg, München, 1991.

[Paulus, 2001] Paulus, D. *Aktives Bildverstehen*. Der Andere Verlag, Osnabrück, Germany, 2001.

[Pösl, 1999] Pösl, J. *Erscheinungsbasierte statistische Objekterkennung*. Berichte aus der Informatik. Shaker Verlag, Aachen, Germany, 1999.

[Rabiner und Juang, 1993] Rabiner, L., Juang, B.-H. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, N.J., 1993.

[Rechenberg, 1973] Rechenberg, I. *Evolutionsstrategie: Optimierung technischer Systeme nach den Prinzipien der biologischen Evolution*. problemata 15. Frommann-Holzboog, Stuttgart, Germany, 1973.

[Rechenberg, 1994] Rechenberg, I. *Evolutionsstrategie '94*, Bd. 1 von *Werkstatt Bionik und Evolutionstechnik*. Frommann-Holzboog, Stuttgart, Germany, 1994.

[Reinhold, 2003] Reinhold, M. *Robuste, probabilistische, erscheinungsbasierte Objekterkennung*. Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 2003.

[Ritter et al., 1991] Ritter, H., Martinetz, T., Schulten, K. *Neuronale Netze*. Addison-Wesley, Bonn, 2. Aufl., 1991.

[Sagerer und Niemann, 1997] Sagerer, G., Niemann, H. *Semantic Networks for Understanding Scenes*. Advances in Computer Vision and Machine Intelligence. Plenum Press, New York and London, 1997.

[Schneeweiß, 1974] Schneeweiß. *Dynamisches Programmieren*. Physica Verlag, Würzburg, Wien, 1974.

[Schölkopf, 1997] Schölkopf, B. *Support Vector Learning*. GMD-Bericht Nr. 287. Oldenbourg, München, 1997.

[Schrimpf et al., 2000] Schrimpf, G., Schneider, J., Stamm-Wildbrandt, H., Dueck, G. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159:139–171, 2000.

[Schukat-Talamazzini, 1995] Schukat-Talamazzini, E.G. *Automatische Spracherkennung*. Vieweg, Wiesbaden, 1995.

[Schürmann, 1977] Schürmann, J. *Polynomklassifikatoren für die Zeichenerkennung*. R. Oldenbourg, München, 1977.

[Scudder, 1965] Scudder, H.J. Adaptive communication receivers. *IEEE Trans. on Information Theory*, 11:167–174, 1965.

[Sebestyen, 1962] Sebestyen, G. *Decision Making Processes in Pattern Recognition*. MacMillan, New York, 1962.

[Sontag und Sussmann, 1985] Sontag, E., Sussmann, H. Image restoration and segmentation using the annealing algorithm. In *Proc. of the 24th Conference on Decision and Control*, S. 768–773. Ft. Lauderdale, 1985.

[Spall, 2003] Spall, J.C. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. J. Wiley, Hoboken, NJ, USA, 2003.

[Toern und Zilinskas, 1989] Toern, A., Zilinskas, A. *Global Optimization*. Lecture Notes in Computer

Science, Nr. 350. Springer, Berlin, 1989.

[Tou und Gonzalez, 1974] Tou, J. T., Gonzalez, R. C. *Pattern Recognition Principles*. Addison-Wesley, New York, 1974.

[Tsypkin, 1973] Tsypkin, Y.Z. *Foundations of the Theory of Learning Systems*. Academic Press, New York, 1973.

[Wasan, 1969] Wasan, M.T. *Stochastic Approximation*. Cambridge University Press, Cambridge, 1969.

[Wells III, 1993] Wells III, W.M. *Statistical Object Recognition*. Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA, Massachusetts, 1993.

[Wolpert und Macready, 1997] Wolpert, D.H., Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. on Evolutionary Computation*, 1(1):67–82, 1997.

[Ye, 1997] Ye, Y. *Interior Point Algorithms: Theory and Analysis*. J. Wiley, New York, 1997.

[Young und Calvert, 1973] Young, T. Y., Calvert, T. W. *Classification, Estimation, and Pattern Recognition*. Elsevier, New York, 1973.

[Zell, 1994] Zell, A. *Simulation Neuronaler Netze*. Addison-Wesley (Deutschland) GmbH, Bonn, 1994.

[Zribi und Ghorbel, 2003] Zribi, M., Ghorbel, F. An unsupervised and non-parametric Bayesian classifier. *Pattern Recognition Letters*, 24:97–112, 2003.

Kapitel 2

Vorverarbeitung (VK.1.3.3, 18.05.2007)

Mit **Vorverarbeitung** werden hier solche Transformationen bezeichnet, die ein vorgegebenes Muster in ein anderes überführen – also z. B. eine Ziffer 3 in eine andere Ziffer 3 – wobei jedoch das transformierte Muster für die weitere Verarbeitung geeigneter sein soll. Das führt sofort auf das Problem, den Erfolg oder den Nutzen von Vorverarbeitungsmaßnahmen konkret zu bewerten. Dieses ist i. Allg. ein äußerst schwieriges Problem, da der Erfolg nicht nur von der eigentlichen Vorverarbeitung sondern auch von den nachfolgenden Operationen abhängt. Man muss also ein vollständiges Klassifikationssystem gemäß Bild 1.4.1, S. 26, realisieren und dessen Leistungsfähigkeit in Abhängigkeit von verschiedenen Vorverarbeitungsoperationen messen. Der Aufwand dafür ist erheblich, er wurde aber durchaus für verschiedene Klassifikationsaufgaben betrieben. Um diesen Aufwand zu vermeiden oder auch um mögliche sinnvolle Transformationen von weniger sinnvollen zu trennen, werden vielfach heuristische Beurteilungskriterien herangezogen. Ein wichtiges Kriterium ist die subjektive Beurteilung der „Qualität“ eines Musters vor und nach der Vorverarbeitung durch Ansehen oder Anhören. Ein weiteres Kriterium ergibt sich aus der intuitiv einleuchtenden Überlegung, dass die Klassifikation von Mustern umso einfacher sein sollte je weniger sich Muster einer Klasse voneinander unterscheiden. Man sollte also versuchen, die Variabilität zwischen den Mustern einer Klasse zu reduzieren. Obwohl die Überlegung einleuchten mag, ist die Reduzierung der Variabilität natürlich nur dann lohnend, wenn der dafür erforderliche Aufwand entweder zu einem entsprechend reduzierten Aufwand bei der nachfolgenden Verarbeitung oder zu einer Erhöhung der Leistungsfähigkeit des Systems führt. Damit ist man wieder beim Test des Gesamtsystems. Es wäre ohne Zweifel ein wichtiger Fortschritt, wenn es gelänge, Vorverarbeitungsmaßnahmen unabhängig vom Gesamtsystem zu bewerten. Zur Zeit ist nicht bekannt, wie das zu tun ist, und es ist nicht einmal bekannt, ob es überhaupt möglich ist. Die Grenze zwischen Vorverarbeitung und der im nächsten Kapitel zu behandelnden Merkmalsgewinnung ist oft nicht völlig eindeutig zu ziehen, und in manchen Veröffentlichungen wird auch die Merkmalsgewinnung als Teil der Vorverarbeitung betrachtet.

In diesem Kapitel werden folgende Gruppen von Operationen für die Vorverarbeitung behandelt.

1. Kodierung – die effektive Darstellung von Mustern in einer für den Digitalrechner geeigneten Form.
2. Schwellwertoperationen – die Auswahl einiger und Unterdrückung der restlichen Funktionswerte.
3. Lineare Operationen – die Beseitigung oder Verbesserung fehlerhafter oder einfach unnötiger Funktionswerte des Musters mit linearen Transformationen.

4. Nichtlineare Operationen – die Verwendung nichtlinearer Transformationen für diesen Zweck.
5. Normierungsmaßnahmen – die Angleichung der Werte einiger Parameter an Normalwerte oder -wertebereiche.
6. Operationen auf diskreten Mustern – einige grundsätzliche Ergebnisse zur Verarbeitung diskreter Muster.

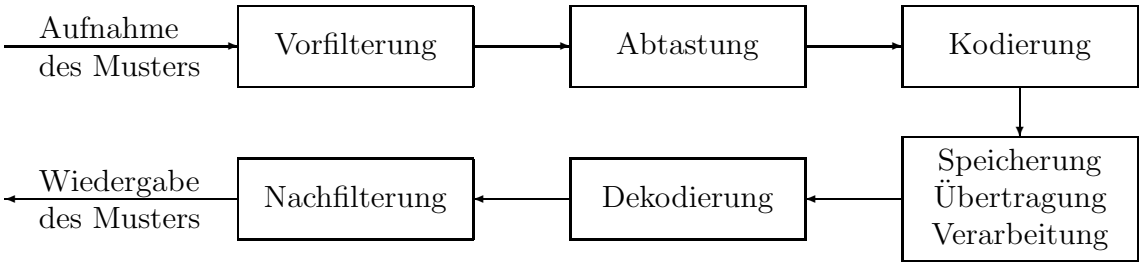


Bild 2.1.1: Aufnahme und Wiedergabe von Mustern mit dazwischenliegender digitaler Verarbeitung

2.1 Kodierung (VA.1.2.3, 18.05.2007)

2.1.1 Allgemeine Bemerkungen

Da ein Muster als Funktion $\boldsymbol{f}(\boldsymbol{x})$ definiert wurde, kann zunächst der Eindruck entstehen, dass man diese Funktion in geschlossener Form, z. B. von der Art $f(x) = ax^2+bx+c$ oder $f(x,y) = \exp[-x^2 - y^2]$ angeben könnte. Das ist natürlich nicht der Fall, da Muster – man betrachte nochmals Bild 1.2.2 und Bild 1.2.3, S. 19, – i. Allg. keine solche Darstellung erlauben. Der einzig gangbare Weg ist die Definition von $\boldsymbol{f}(\boldsymbol{x})$ durch eine *Wertetabelle*, d. h. man ermittelt und speichert die Funktionswerte $\boldsymbol{f}$ für eine endliche Zahl M von Koordinatenwerten $\boldsymbol{x}_i$, $i = 0, 1, \ldots, M-1$; dieser Vorgang wird als *Abtastung* ("sampling") von $\boldsymbol{f}(\boldsymbol{x})$ bezeichnet. Die Verarbeitung der Funktionswerte $\boldsymbol{f}$ erfolgt vielfach digital, und in diesem Buch werden nur digitale Verfahren behandelt. Das bedeutet, dass auch die Funktionswerte nur mit endlich vielen diskreten Quantisierungsstufen ermittelt, gespeichert und verarbeitet werden; der Vorgang der Zuordnung diskreter Funktionswerte wird als *Kodierung* bezeichnet. Damit wird der Tatsache Rechnung getragen, dass Digitalrechner nur endlich viele Werte speichern und in endlicher Zeit nur endlich viele bit verarbeiten können.

Damit ergibt sich die in Bild 2.1.1 gezeigte Folge von Schritten. Ein Muster wird zunächst aufgenommen, also eine physikalische Größe wie Schalldruck oder Lichtintensität in eine elektrische Spannung umgewandelt. Es folgt eine Vorfilterung, um die noch zu erörternde Bedingung (2.1.18) sicherzustellen, und die Abtastung des Musters. Die an diskreten Koordinatenwerten gemessenen Funktionswerte werden kodiert. Die Funktion $\boldsymbol{f}(\boldsymbol{x})$ ist damit durch endlich viele diskrete Werte dargestellt und kann in einem Digitalrechner gespeichert und verarbeitet werden. Soll das gespeicherte Muster wieder dargestellt werden, müssen die kodierten Funktionswerte dekodiert und die abgetasteten Koordinatenwerte durch Interpolation ergänzt werden; letzteres kann z. B. durch ein Filter erfolgen. Für die digitale Verarbeitung von Mustern sind also die folgenden grundsätzlichen Fragen zu klären.

1. Wieviele Abtastwerte braucht man zur angemessenen Darstellung einer Funktion, bzw. wie groß muss das *Abtastintervall* sein?
2. Wieviele *Quantisierungsstufen* braucht man zur angemessenen Darstellung der Funktionswerte?
3. Wie sind die Stufen zu wählen, d. h. wie muss die *Quantisierungskennlinie* aussehen?

Diese Fragen werden in den nächsten beiden Abschnitten erörtert. Ohne auf Einzelheiten einzugehen sei noch erwähnt, dass je nach Art des Aufnahme– oder Wiedergabegerätes einige der

Schritte Vorfilterung, Abtastung, Nachfilterung in diesen Geräten direkt durchgeführt werden können. Es kann auch sein, dass physiologische Eigenheiten der menschlichen Sinnesorgane ausgenutzt werden, wie es z. B. bei der Darstellung bewegter Bilder durch eine genügend schnelle Folge statischer Bilder üblich ist.

2.1.2 Abtastung

Vorgehensweise

Im Folgenden wird stets vorausgesetzt, dass die **Abtastung** einer Funktion an *äquidistanten Stützstellen* erfolgt. Beispielsweise bedeutet das für ein Grauwertbild $f(x, y)$, dass es durch eine Bildmatrix $\boldsymbol{f}$ (Wertetabelle) ersetzt wird gemäß

$$\begin{aligned} f(x,y) \quad &\Longrightarrow \quad f_{jk}\ , \qquad \boldsymbol{f} = [f_{jk}]\ , \\ f_{jk} \quad &= \quad f(x_0 + j\Delta x, y_0 + k\Delta y)\ , \\ &\qquad j = 0, 1, \ldots, M_x - 1\ , \qquad k = 0, 1, \ldots, M_y - 1\ . \end{aligned} \tag{2.1.1}$$

Dabei sind x_0, y_0 beliebige Anfangskoordinaten, Δx und Δy sind die Abstände der Stützstellen, und die Endkoordinaten sind

$$\begin{aligned} x_1 &= x_0 + (M_x - 1)\Delta x\ , \\ y_1 &= y_0 + (M_y - 1)\Delta y\ . \end{aligned} \tag{2.1.2}$$

Im weiteren Text wird mit f_{jk} stets ein einzelner **Abtastwert** bezeichnet, mit $[f_{jk}]$ eine Folge von Abtastwerten einer zweidimensionalen Funktion, wobei aus dem Kontext hervorgeht, ob diese Folge endlich oder unendlich ist, und mit $\boldsymbol{f}$ ganz allgemein eine diskrete Darstellung einer ein– oder mehrdimensionalen Funktion durch Abtastwerte. Offensichtlich reicht bei bekannten $x_0, y_0, \Delta x, \Delta y$ die Angabe von f_{jk} zur eindeutigen Kennzeichnung eines Abtastwertes aus. Man kann zur Vereinfachung ohne Beschränkung der Allgemeinheit $x_0 = y_0 = 0$ und $\Delta x = 1$ Längeneinheit in x–Richtung, $\Delta y = 1$ Längeneinheit in y–Richtung setzen. In diesem Falle ist einfach

$$f_{jk} = f(j, k)\ , \qquad j = 0, 1, \ldots, M_x - 1\ ; \quad k = 0, 1, \ldots, M_y - 1\ . \tag{2.1.3}$$

Die mit (2.1.1) – (2.1.3) definierte Abtastung lässt sich in offensichtlicher Weise auf Funktionen gemäß (1.2.5), S. 13, mit beliebigen Werten von m und n ausdehnen. Wie aus Bild 2.1.2 hervorgeht, ergibt sich bei rechtwinkligen Koordinaten ein rechteckiges Raster von Abtastpunkten in der Ebene, bei geeignet gewählten schiefwinkligen ein sechseckiges oder hexagonales Raster. Hier werden, falls nichts anderes ausdrücklich erwähnt wird, stets rechteckige Koordinaten verwendet.

Der Vollständigkeit halber sind in Bild 2.1.2 unten auch die drei möglichen Aufteilungen einer Ebene mit regelmäßigen Vielecken dargestellt. Es sind dies die Aufteilung mit Quadraten (quadratisches Raster), mit gleichseitigen Sechsecken (hexagonales Raster) und mit gleichseitigen Dreiecken, wobei letztere Aufteilung jedoch in der Mustererkennung keine praktische Bedeutung hat. Für einen Rasterpunkt P sind jeweils die möglichen Nachbarn gezeigt, wobei man unter Nachbarn entweder solche Punkte versteht, die mit P eine gemeinsame Seite haben, oder solche, die mit P eine gemeinsame Seite oder eine gemeinsame Ecke haben. Der Vorteil des hexagonalen Rasters besteht darin, dass es nur einen Typ von Nachbarn gibt, der des quadratischen, dass übliche Abtastgeräte im quadratischen Raster arbeiten. Die im quadratischen Raster üblichen 4– und 8–Nachbarschaften sind in Abschnitt 2.6.1 definiert.

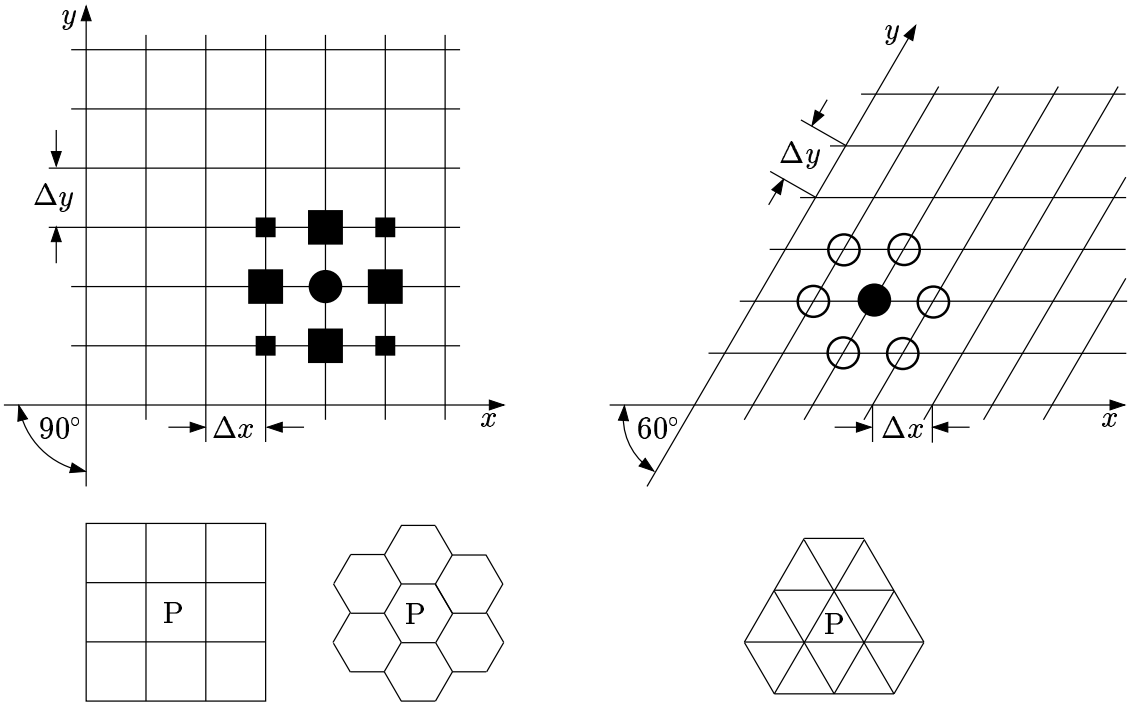


Bild 2.1.2: Rechteckiges und hexagonales Abtastraster. In ersterem hat ein Punkt entweder vier Nachbarn in gleichem Abstand oder acht in unterschiedlichem Abstand, je nach Definition der Nachbarschaft. In letzterem hat ein Punkt sechs Nachbarn in gleichem Abstand. Darunter die drei regelmäßigen Zerlegungen der Ebene

Das *Abtasttheorem* ermöglicht eine Aussage über den erforderlichen Abstand $\Delta x, \Delta y$ der Abtastpunkte. Zusammen mit (2.1.2) ergibt sich bei bekannten Anfangs– und Endkoordinaten daraus die erforderliche Anzahl der Abtastpunkte M_x, M_y. Der Einfachheit halber wird es hier nur für eine Funktion $f(x)$ von einer Variablen angegeben, jedoch lässt es sich ohne weiteres verallgemeinern.

Eindimensionale FOURIER-Transformation

Da im Folgenden die FOURIER-Transformation und die FOURIER-Reihe gebraucht werden, wird an die relevanten Gleichungen kurz erinnert. Die zweidimensionale FOURIER-Transformation wird in Definition 2.6, S. 92 vorgestellt.

Definition 2.1 *Die (eindimensionale)* FOURIER-**Transformierte** $F(\xi)$ *einer Funktion* $f(x)$ *mit* $\int_{-\infty}^{\infty} |f(x)|\,\mathrm{d}x < \infty$ *ist definiert durch*

$$F(\xi) \;=\; \int_{-\infty}^{\infty} f(x)\exp[-\mathrm{i}\,\xi x]\;\mathrm{d}x \;=\; \mathrm{FT}\{f(x)\}\;. \tag{2.1.4}$$

Aus $F(\xi)$ *erhält man* $f(x)$ *aus dem* **Umkehrintegral**

$$f(x) \;=\; \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\xi)\exp[\mathrm{i}\,\xi x]\;\mathrm{d}\xi \;=\; \mathrm{FT}^{-1}\{F(\xi)\}\;. \tag{2.1.5}$$

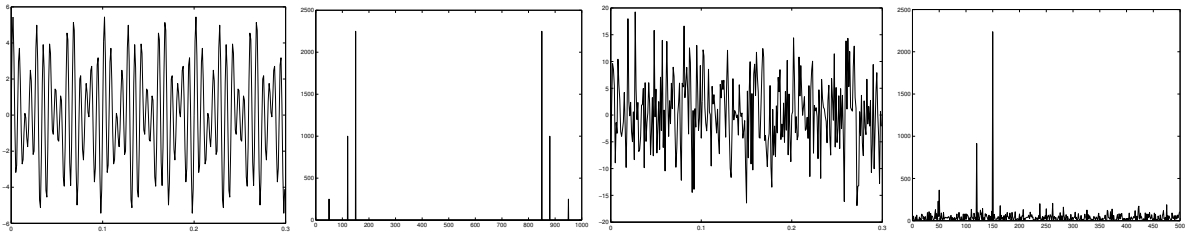


Bild 2.1.3: Das Bild zeigt (von links nach rechts) eine Funktion, die durch Überlagerung von drei Sinusfunktionen entstand; den Betrag der FOURIER-Transformierten; die Funktion mit zusätzlichem Rauschen überlagert; deren FOURIER-Transformierte halber Länge

Das Symbol $\mathfrak{i}$ (hier im Exponenten einer e–Funktion) ist stets die *komplexe* Zahl $(0,1)$ mit $\mathfrak{i}^2 = (-1,0) = -1$ bzw. $\sqrt{-1} = \mathfrak{i}$. Bild 2.1.3 zeigt ein Beispiel für Funktionen und ihre FOURIER-Transformierten. Man sieht, dass in beiden Fällen die drei Frequenzen der transformierten Funktion (50, 120 und 150 Hz) als Maxima erkennbar sind; man sieht auch, dass es wegen der Symmetrie reicht, nur die Hälfte der Werte zu berechnen. Das FOURIER-Integral wurde durch die in Abschnitt 2.3.3 eingeführte diskrete Version approximiert.

Einige Beispiele für Transformationspaare sind

$$f(x) = 1\,, \qquad F(\xi) = 2\pi\delta(\xi)\,, \tag{2.1.6}$$

$$f(x) = \begin{cases} 1 & : \ |x| < x_e \\ 0 & : \ \text{sonst} \end{cases}\,, \qquad F(\xi) = \frac{2\,\sin[x_e\,\xi]}{\xi}\,, \tag{2.1.7}$$

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]\,, \quad F(\xi) = \exp\left[\mathfrak{i}\,\mu\xi - \frac{1}{2}\sigma^2\xi^2\right] \tag{2.1.8}$$
$$= \mathcal{N}(x\,|\,\mu,\sigma)\,.$$

Dabei ist die **Delta–Funktion** in (2.1.6) definiert durch

$$\delta(x) = \lim_{\sigma\to 0}\frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\frac{x^2}{2\sigma^2}\right]\,, \quad \int_{-\infty}^{\infty}\delta(x)\,\mathrm{d}x = 1\,. \tag{2.1.9}$$

Die zweidimensionale Version von (2.1.7) zeigt Bild 2.3.4, S. 95, mitte. Die Funktion in (2.1.8) wird als GAUSS-**Funktion** oder *Normalverteilung* bezeichnet und ist gezeigt in Bild 4.2.1, S. 324.

Wenn man die FOURIER-Transformierte $F(\xi)$ einer Funktion $f(x)$ kennt, kann man daraus die FOURIER-Transformierten verschiedener Funktionen berechnen, die daraus hervorgehen. Einige Beispiele geben wir ohne Beweis an:

$$f_1(x) = f(x-x_0)\,, \qquad \mathrm{FT}\{f_1(x)\} = F(\xi)\mathrm{exp}[-\mathfrak{i}\,x_0\xi]\,, \tag{2.1.10}$$

$$f_2(x) = \frac{\mathrm{d}^n f(x)}{\mathrm{d}x^n}\,, \qquad \mathrm{FT}\{f_2(x)\} = (\mathfrak{i}\,\xi)^n F(\xi)\,, \tag{2.1.11}$$

$$f_3(x) = f\left(\frac{x}{\sigma}\right)\,, \qquad \mathrm{FT}\{f_3(x)\} = |\sigma| f(\sigma\,x)\,, \tag{2.1.12}$$

$$E = \int_{-\infty}^{\infty}|f(x)|^2\,\mathrm{d}x\,, \qquad E = \frac{1}{2\pi}\int_{-\infty}^{\infty}|F(\xi)|^2\,\mathrm{d}\xi\,, \tag{2.1.13}$$

$$r(x) = \int_{-\infty}^{\infty} f(u)f(u-x)\,\mathrm{d}u\,, \qquad \mathrm{FT}\{r(x)\} = |F(\xi)|^2\,, \tag{2.1.14}$$

$$h(x) \;=\; \int_{-\infty}^{\infty} f(u)g(x-u)\,\mathrm{d}u\;, \qquad \mathrm{FT}\{h(x)\} = F(\xi)G(\xi)\;, \tag{2.1.15}$$

wobei $G(\xi) = \mathrm{FT}\{g(x)\}$ ist. Die Beziehungen (2.1.10) – (2.1.12) sind leicht zu verifizieren. (2.1.13) ist das PARSEVAL-*Theorem*. Die Funktion $r(x)$ in (2.1.14) ist die **Autokorrelationsfunktion** von $f(x)$; ihre FOURIER-Transformierte ist also gleich dem Betragsquadrat der FOURIER-Transformierten von $f(x)$. Schließlich gibt der *Multiplikationssatz* in (2.1.15) eine wichtige Beziehung zwischen dem Ergebnis $h(x)$ der *Faltung* (s. Abschnitt 2.3.2) zweier Funktion $f(x)$, $g(x)$ und deren FOURIER-Transformierten $F(\xi)$, $G(\xi)$; dieser Satz wird wegen seiner Bedeutung in Satz 2.12 für den diskreten Fall genauer betrachtet.

Eine in einem Intervall $(-\xi_0, \xi_0)$ *periodische* Funktion kann man in diesem Intervall in eine FOURIER-Reihe entwickeln.

Definition 2.2 *Eine periodische Funktion $F(\xi)$ hat die* FOURIER-**Reihe**

$$F(\xi) = \sum_{j=-\infty}^{\infty} a_j \exp\left[\mathrm{i}\,2\pi\frac{j\xi}{2\xi_0}\right] \tag{2.1.16}$$

mit den **Entwicklungskoeffizienten**

$$a_j = \frac{1}{2\xi_0}\int_{-\xi_0}^{\xi_0} F(\xi)\exp\left[-\mathrm{i}\,2\pi\frac{j\xi}{2\xi_0}\right]\,\mathrm{d}\xi\;. \tag{2.1.17}$$

Abtasttheorem

Es wird nun *vorausgesetzt*, dass für die abzutastende Funktion $f(x)$ die FOURIER-Transformierte $F(\xi)$ gemäß (2.1.4) existiert und **bandbegrenzt** im Frequenzbereich $(-B_x, B_x)$ ist, d. h. es gilt

$$F(\xi) = 0 \quad \text{für} \quad |\xi| > \xi_0 = 2\pi B_x\;. \tag{2.1.18}$$

Satz 2.1 (**Abtasttheorem**) *Eine bandbegrenzte Funktion $f(x)$ ist vollständig bestimmt durch die Abtastwerte*

$$f_j = f(j\Delta x)\;, \tag{2.1.19}$$

wenn man als Abstand der Abtastwerte

$$\Delta x \le \frac{1}{2B_x} = \frac{\pi}{\xi_0} \tag{2.1.20}$$

wählt. Man kann $f(x)$ rekonstruieren aus der Interpolationsformel

$$f(x) = \sum_{j=-\infty}^{\infty} f_j\,\frac{\sin\left[2\pi B_x(x-j\Delta x)\right]}{2\pi B_x(x-j\Delta x)}\;. \tag{2.1.21}$$

Setzt man, wie oben beim Übergang von (2.1.1) auf (2.1.3) $\Delta x = 1$, so folgt daraus mit (2.1.20) $2B_x = 1$, und damit reduziert sich die Interpolationsformel auf die einfachere Form

$$f(x) = \sum_{j=-\infty}^{\infty} f_j\,\frac{\sin\left[\pi(x-j)\right]}{\pi(x-j)} = \sum_{j=-\infty}^{\infty} f_j\,\mathrm{sinc}[\pi(x-j)]\;. \tag{2.1.22}$$

Beweis: Der hier gebrachte Beweis des Abtasttheorems beruht auf den beiden *Beweisideen*

1. man berechne $f(x)$ aus dem Umkehrintegral der FT und
2. man betrachte $F(\xi)$ als *eine Periode* einer gedanklich periodisch fortgesetzten Funktion *und* entwickle $F(\xi)$ in eine FOURIER-Reihe.

Aus der FOURIER-Transformierten $F(\xi)$ in (2.1.4) erhält man wegen (2.1.18) die Funktion $f(x)$ aus dem Umkehrintegral

$$\begin{aligned} f(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\xi)\exp[\mathrm{i}\,\xi x]\;\mathrm{d}\xi \\ &= \frac{1}{2\pi}\int_{-\xi_0}^{\xi_0} F(\xi)\exp[\mathrm{i}\,\xi x]\;\mathrm{d}\xi\;. \end{aligned} \tag{2.1.23}$$

Da $F(\xi)$ bandbegrenzt im Intervall $(-\xi_0, \xi_0)$ ist, kann man es in diesem Intervall gemäß (2.1.16) in eine FOURIER-Reihe entwickeln, wobei man gedanklich $F(\xi)$ über das Intervall $(-\xi_0, \xi_0)$ hinaus periodisch fortsetzt. Unter Beachtung von (2.1.5) und (2.1.20) erhält man für die Entwicklungskoeffizienten

$$\begin{aligned} a_j &= \frac{1}{2\xi_0}\int_{-\xi_0}^{\xi_0} F(\xi)\exp\left[\frac{-\mathrm{i}\,j2\pi\xi}{2\xi_0}\right]\;\mathrm{d}\xi \\ &= \frac{1}{2\pi}\;\;\frac{\pi}{\xi_0}\int_{-\xi_0}^{\xi_0} F(\xi)\exp\left[\mathrm{i}\,\xi\frac{-j\pi}{\xi_0}\right]\;\mathrm{d}\xi \\ &= f\left(\frac{-j\pi}{\xi_0}\right)\frac{\pi}{\xi_0} \\ &= f(-j\Delta x)\Delta x\;. \end{aligned} \tag{2.1.24}$$

Setzt man (2.1.24) in (2.1.16) ein, so ergibt sich für $F(\xi)$ die Gleichung

$$\begin{aligned} F(\xi) &= \sum_{j=-\infty}^{\infty} f(-j\Delta x)\exp[\mathrm{i}\,j\Delta x\xi]\,\Delta x \\ &= \sum_{j=-\infty}^{\infty} f(j\Delta x)\exp[-\mathrm{i}\,j\Delta x\xi]\,\Delta x\;. \end{aligned}$$

Dieser Ausdruck für $F(\xi)$ in (2.1.5) eingesetzt ergibt schließlich

$$\begin{aligned} f(x) &= \frac{1}{2\pi}\sum_{j=-\infty}^{\infty} f(j\Delta x)\int_{-\xi_0}^{\xi_0}\exp[\mathrm{i}\,\xi(x-j\Delta x)]\,\Delta x\,\mathrm{d}\xi \\ &= \sum_{j=-\infty}^{\infty} f(j\Delta x)\frac{\Delta x}{2\pi}\left[\frac{\exp[\mathrm{i}\,\xi(x-j\Delta x)]}{\mathrm{i}\,(x-j\Delta x)}\right]_{-\xi_0}^{\xi_0} \\ &= \sum_{j=-\infty}^{\infty} f_j\frac{\sin\left[2\pi B_x(x-j\Delta x)\right]}{2\pi B_x(x-j\Delta x)}\;. \end{aligned}$$

Die letzte Gleichung ist gerade (2.1.21), und damit ist Satz 2.1 bewiesen.

Der obige Satz ist die theoretische Grundlage für die digitale Verarbeitung von Mustern, da er sicherstellt, dass man ein Muster unter bestimmten Voraussetzungen durch seine Abtastwerte nicht nur approximieren, sondern sogar *exakt* darstellen kann. Ein Beispiel der Approximati-

on zeigt Bild 2.1.5. Allerdings ist bei Mustern, die praktisch immer auf ein endliches Intervall $(x_0 \leq x \leq x_1)$ beschränkt sind, die Bandbegrenzung gemäß (2.1.18) nie genau eingehalten. Der Grund liegt in den folgenden beiden Sätzen, die hier ohne Beweis angegeben werden. Es sei wieder $f(x)$ eine Funktion mit der FOURIER-Transformierten $F(\xi)$. Definiert man die *Bandbreite* $\Delta\xi^2$ und die *Ortsspreizung* (bzw. Zeitspreizung) Δx^2 mit

$$\Delta\xi^2 = \frac{\int \xi^2 |F(\xi)|^2 \,\mathrm{d}\xi}{\int |F(\xi)|^2 \,\mathrm{d}\xi} , \quad \Delta x^2 = \frac{\int x^2 |f(x)|^2 \,\mathrm{d}x}{\int |f(x)|^2 \,\mathrm{d}x} ,$$

so gilt der folgende Satz:

Satz 2.2 *Für das* Zeit-Bandbreite-Produkt *gilt die* **Unschärferelation**

$$\Delta x \Delta\, \xi \geq \frac{1}{4\pi} . \tag{2.1.25}$$

Für Funktionen von zwei Variablen gelten zwei analoge Beziehungen für die beiden Orts– (bzw. Zeit–) und Frequenzvariablen. Ohne Beweis wird erwähnt, dass die GAUSS-Funktion (2.1.8) die *einzige reellwertige* Funktion ist, für die das Gleichheitszeichen gilt, und die unten in Definition 3.10, S. 196, eingeführte GABOR-Funktion die *einzige komplexwertige* Funktion ist, für die das Gleichheitszeichen gilt. Diese Funktionen erreichen die bestmögliche Konzentration sowohl im Orts– (bzw. Zeit–) als auch im Frequenzbereich. Eine Funktion kann also *nicht gleichzeitig* im Orts– (bzw. Zeit–) *und* im Frequenzbereich beliebig konzentriert sein, wie Bild 2.1.4 zeigt. Diese Aussage trifft auch der folgende Satz:

Satz 2.3 *Es gibt keine Funktion (in L_2), die* sowohl *bandbegrenzt* als auch *ortsbegrenzt (bzw. zeitbegrenzt) ist (außer der identisch verschwindenden Funktion).*
Beweis*: s. z. B. [Paley und Wiener, 1934].*

Um diesen Problemen zu begegnen, muss man, wie in Bild 2.1.1 angedeutet, vor der Abtastung die Einhaltung von (2.1.18) durch eine Vorfilterung, bei der Frequenzen $|\xi| > \xi_0$ möglichst gut unterdrückt werden, sicherstellen. Tatsächlich wird man also i. Allg. Muster $f(x)$ durch Abtastung umso genauer approximieren je größer der vom Filter durchgelassene Frequenzbereich ist, das heißt aber wegen (2.1.20) auch je kleiner die Abtastschrittweite und damit je größer die Zahl der Abtastpunkte ist. Da man diese Zahl zur Beschränkung des Verarbeitungs- und Speicheraufwandes klein halten möchte, muss hier ein Kompromiss zwischen Aufwand und Genauigkeit geschlossen werden. Zu den Ungenauigkeiten, die durch die Bandbegrenzung und Abtastung verursacht werden, kommen die im nächsten Abschnitt diskutierten, durch die Quantisierung verursachten, hinzu. Bei mehrdimensionalen Mustern, z. B. $f(x, y)$, sind (2.1.18), (2.1.20) in der naheliegenden Weise zu verallgemeinern, dass man

$$F(\xi, \eta) = 0 \qquad \text{für} \qquad \begin{matrix} |\xi| > \xi_0 = 2\pi B_x , \\ |\eta| > \eta_0 = 2\pi B_y \end{matrix} \tag{2.1.26}$$

$$\Delta x \leq \frac{1}{2\pi B_x} \qquad \text{und} \qquad \Delta y \leq \frac{1}{2\pi B_y} \tag{2.1.27}$$

fordert.

Bezeichnet man mit $1/\Delta x$ die Zahl der Abtastpunkte je Längen– oder Zeiteinheit, die auch als *Abtastfrequenz* bezeichnet wird, so ist (2.1.20) gleichwertig der Aussage, dass die Abtastfre-

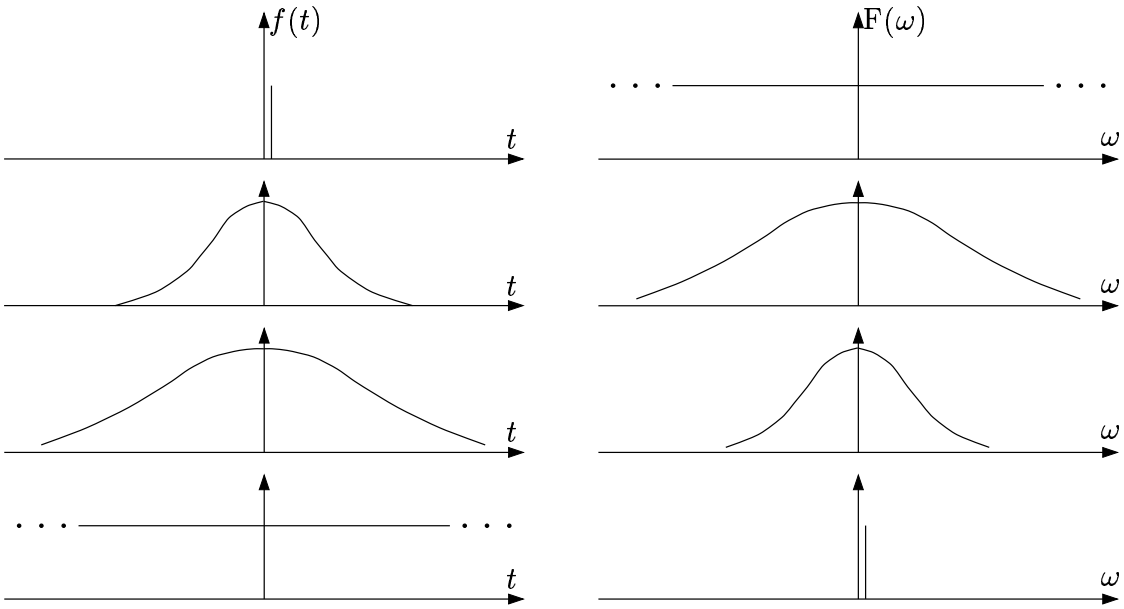


Bild 2.1.4: Je schmaler eine Funktion im Zeit– bzw. Ortsbereich ist, um so breiter ist sie im Frequenzbereich, und umgekehrt

quenz mindestens gleich der doppelten Grenzfrequenz B_x sein muss. Schließlich wird erwähnt, dass die Abtastung an einem Zeit*punkt* eine Idealisierung ist, die praktisch nicht realisierbar ist; physikalisch realisierte Abtastungseinheiten messen stets den Funktionswert in einem kleinen Orts– bzw. Zeit*intervall*. Dieser Gesichtspunkt wurde oben vernachlässigt. In Abschnitt 3.3.3 wird angemerkt, dass eine Abtastung mit einem endlichen Intervall (z. B. durch die Rezeptoren einer CCD–Kamera) als Entwicklung der kontinuierlichen Funktion $f(t)$ nach geeigneten Skalierungsfunktionen aufgefasst werden kann.

2.1.3 Puls Kode Modulation

Im Allgemeinen können die Abtastwerte f_j irgendeinen Wert aus dem kontinuierlichen Wertebereich $f_{\min} \leq f_j \leq f_{\max}$ annehmen. Für die digitale Verarbeitung muss auch der Wertebereich quantisiert werden, wobei es zweckmäßig ist, $L = 2^B$ Stufen zu wählen, die durch die ganzen Zahlen $0, 1, \ldots, 2^B - 1$ kodiert und in B bit eines Rechners gespeichert werden. Wenn $\boldsymbol{f}(\boldsymbol{x})$ eine vektorwertige Funktion ist, wird dieses Verfahren auf jede Komponente angewendet. Das Prinzip der Abbildung von Abtastwerten f_j in diskrete Werte f'_j zeigt Bild 2.1.6. Abtastung des Musters an diskreten Koordinatenwerten und Quantisierung der Abtastwerte in diskrete Amplitudenstufen ergeben die *Puls Kode Modulation* (PCM), die ein einfaches und grundlegendes Kodierverfahren ist.

Da die Wahl des Abtastintervalls im vorigen Abschnitt erörtert wurde, bleiben hier noch zwei Fragen zu klären, nämlich wieviele Quantisierungsstufen zu wählen sind und wie die Quantisierungskennlinie aussehen sollte – die in Bild 2.1.6 gezeigte lineare Kennlinie ist ja keineswegs die einzig mögliche. Die Zahl der Quantisierungsstufen bestimmt die Genauigkeit, mit der f_j durch f'_j approximiert wird. Definiert man den Quantisierungsfehler oder das *Quantisierungsrauschen* mit

$$n_j = f_j - f'_j \tag{2.1.28}$$

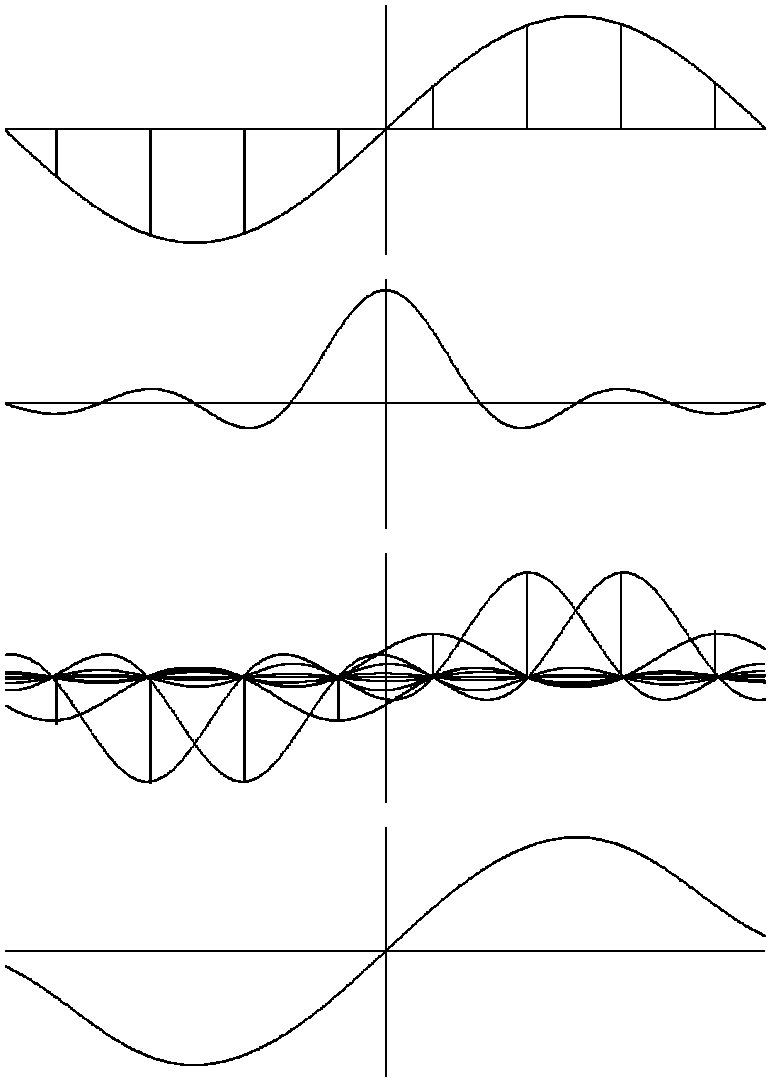


Bild 2.1.5: Zur Veranschaulichung von Satz 2.1 ist von oben nach unten Folgendes dargestellt: a) Eine Funktion $f(x)$ mit Abtastwerten f_j, letztere angedeutet durch senkrechte Striche. b) Die in (2.1.21) auftretende Funktion $(\sin x)/x$. c) Die zu den Abtastwerten von a) gehörigen Summanden in (2.1.21). d) Die mit (2.1.21) rekonstruierte Funktion, die aus den oben diskutierten Gründen nicht exakt mit der in a) gegebenen Funktion übereinstimmt.

so ist das Verhältnis von Signalenergie zu Rauschenergie ("signal-to-noise-ratio", SNR)

$$\mathrm{SNR} = r' = \frac{E\{f_j^2\}}{E\{n_j^2\}} \tag{2.1.29}$$

ein mögliches Maß für die Genauigkeit der PCM. In (2.1.29) ist $E\{.\}$ der Erwartungswert der in Klammern stehenden Größe. Damit gilt

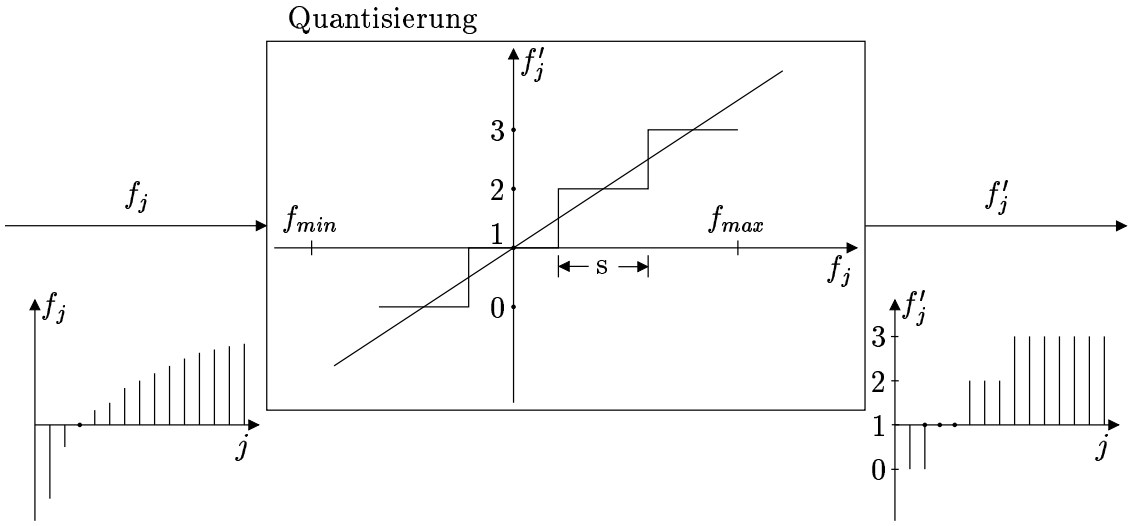


Bild 2.1.6: Quantisierung der Abtastwerte f_j

Satz 2.4 *Unter den im Beweis genannten Voraussetzungen und mit*

$$r = 10 \log r' \tag{2.1.30}$$

gilt die Beziehung

$$r = 6B - 7{,}2\ . \tag{2.1.31}$$

Beweis: Es wird angenommen, dass $E\{f\} = E\{n\} = 0$ ist. Wenn die Zahl der Quantisierungsstufen genügend groß ist, etwa $B > 6$, ist die Annahme eines gleichverteilten Quantisierungsfehlers n berechtigt. Wenn s die in Bild 2.1.6 gezeigte Schrittweite des Quantisierers ist, so ist die Verteilungsdichte des Fehlers

$$p(n) = \begin{cases} \frac{1}{s} & : \quad \frac{-s}{2} \le n \le \frac{s}{2} \quad , \\ 0 & : \quad \text{sonst} \qquad\qquad . \end{cases} \tag{2.1.32}$$

Dabei ist vorausgesetzt, dass der Quantisierer nicht übersteuert wird. Damit erhält man für die Varianz des Fehlers

$$E\{n_j^2\} = \int_{\frac{-s}{2}}^{\frac{s}{2}} \frac{1}{s} n^2 \, \mathrm{d}n = \frac{s^2}{12}\ . \tag{2.1.33}$$

Mit der Bezeichnung

$$\sigma_f = \sqrt{E\{f_j^2\}} \tag{2.1.34}$$

und der Annahme

$$f_{\min} = -4\sigma_f\ , \qquad f_{\max} = 4\sigma_f \tag{2.1.35}$$

erhält man als Schrittweite

$$s = \frac{8\sigma_f}{2^B}\ . \tag{2.1.36}$$

Die Annahme (2.1.35) ist problematisch, da es Signale geben kann, die ihr nicht genügen. Setzt man (2.1.33), (2.1.34), (2.1.36) in (2.1.29) ein, erhält man

$$r' = 12 \cdot 2^{2B-6} \quad . \tag{2.1.37}$$

Zusammen mit (2.1.30) ergibt sich daraus (2.1.31), sodass der Beweis von Satz 2.4 vollständig ist.

Aus (2.1.31) folgt, dass ein bit mehr oder weniger eine Erniedrigung oder Erhöhung des Quantisierungsfehlers um 6dB bedeutet. Diese Aussage gibt zwar einen ersten quantitativen Eindruck vom Einfluss der Quantisierungsstufen auf die Genauigkeit der Darstellung. Sie sagt aber wenig darüber aus, wie viele Stufen oder bit man tatsächlich nehmen sollte. Dafür ist eine genaue Untersuchung der Verarbeitungskette gemäß Bild 1.4.1 oder Bild 2.1.1 – je nach Anwendungsfall – erforderlich. In Bild 2.1.1 ist am Schluss ausdrücklich die Wiedergabe der Muster erwähnt, also die Darstellung für einen menschlichen Beobachter. In diesem Falle wird die Zahl der Quantisierungsstufen so gewählt, dass der subjektive Eindruck des Beobachters, z. B. beim Anhören von Sprache oder Ansehen eines Bildes, zufriedenstellend ist. Letzterer Begriff ist sehr dehnbar, da „zufriedenstellend“ bei Sprache die Verständlichkeit sein kann oder auch die subjektiv als verzerrungsfrei empfundene Wiedergabe. Grundsätzlich ist die Quantisierung der Amplitudenstufen deshalb möglich, weil ein Mensch zwei Sinneseindrücke – gleichgültig ob Druck, Helligkeit, Lautstärke usw. – nur dann subjektiv unterscheiden kann, wenn ihre Intensitäten sich um einen bestimmten Mindestwert unterscheiden (WEBER–FECHNER-Gesetz). Erfahrungsgemäß gelten bei Sprache 11 bit, bei Grauwertbildern 8 bit und bei Farbbildern 8 bit je Farbkanal als ausreichend für gute subjektive Qualität bei der Wiedergabe. Zum Beispiel gilt dann bei den quantisierten Grauwertbildern $f_{jk} \in \{0, 1, \ldots, 255\}$. In Bild 1.4.1 kommt es nicht auf die Wiedergabe, sondern die Klassifikation eines Musters an. Es fehlen systematische Untersuchungen über den Einfluss der Zahl der Quantisierungsstufen auf die Klassifikatorleistung. Meistens orientiert man sich daher bei der Wahl der Stufenzahl ebenfalls am subjektiven Eindruck eines Beobachters.

Es ist naheliegend, eine **Quantisierungskennlinie** zu suchen, die ein definiertes Gütekriterium optimiert. Ein mögliches Kriterium ist der mittlere quadratische Fehler

$$\varepsilon = \sum_{\nu=1}^{L} \int_{a_\nu}^{a_{\nu+1}} (f - b_\nu)^2 p(f)\, \mathrm{d}f \ , \tag{2.1.38}$$

wobei sich die Bezeichnungen aus Bild 2.1.7 ergeben. Alle Werte $a_\nu \leq f_j < a_{\nu+1}$ werden also durch den quantisierten Wert b_ν dargestellt. Nach der obigen Diskussion wäre zwar für die Wiedergabe ein Kriterium, das den subjektiven Fehlereindruck des Beobachters oder die Leistung eines Klassifikationssystems bewertet, vorzuziehen. Wegen der einfacheren mathematischen Behandlung wird hier aber nur (2.1.38) verwendet. Die optimale Quantisierungskennlinie ist durch die Werte a_ν, b_ν gekennzeichnet, für die der Fehler ε in (2.1.38) minimiert wird.

Satz 2.5 *Die optimalen Werte a_ν, b_ν, welche (2.1.38) minimieren, sind gegeben durch*

$$a_\nu = \frac{b_{\nu-1} + b_\nu}{2} \ , \qquad \nu = 2, 3, \ldots, L = 2^B \ , \tag{2.1.39}$$

$$b_\nu = \frac{\int_{a_\nu}^{a_{\nu+1}} f p(f)\, \mathrm{d}f}{\int_{a_\nu}^{a_{\nu+1}} p(f)\, \mathrm{d}f} \ , \quad \nu = 1, \ldots, L \ . \tag{2.1.40}$$

Dabei ist $p(f = a_\nu) \neq 0$ vorausgesetzt.

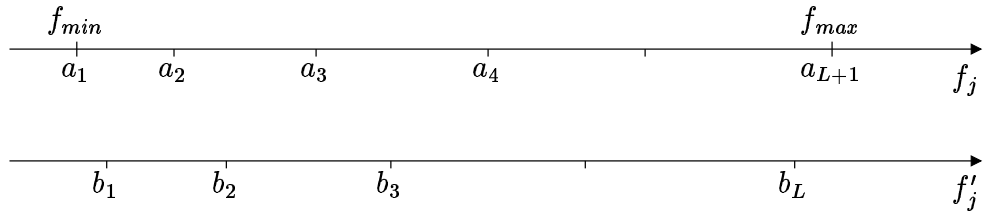


Bild 2.1.7: Zur Bestimmung einer optimalen Quantisierungskennlinie

Beweis: Die Bildung der partiellen Ableitung von (2.1.38) nach b_ν und Nullsetzen derselben ergibt

$$\frac{\partial f}{\partial b_\nu} = \sum_{\nu=1}^{L} \int_{a_\nu}^{a_{\nu+1}} -2(f - b_\nu)p(f)\,\mathrm{d}f = 0 \; .$$

Daraus folgt unmittelbar (2.1.40). Diese Vorgehensweise ergibt für a_ν

$$\frac{\partial f}{\partial a_\nu} = \sum_{\nu=2}^{L} (a_\nu - b_{\nu-1})^2 p(a_\nu) - (a_\nu - b_\nu)^2 p(a_\nu) = 0 \; .$$

Die Werte a_1 und a_{L+1} sind gemäß Bild 2.1.7 festgelegt. Wenn $p(a_\nu) \neq 0$ ist, ist obige Gleichung erfüllt, wenn

$$(a_\nu - b_{\nu-1})^2 = (a_\nu - b_\nu)^2$$

gilt. Daraus folgt sofort (2.1.39), und damit ist Satz 2.5 bewiesen.

Der obige Satz zeigt, dass i. Allg. die Lage der a_ν, b_ν und damit die Quantisierungskennlinie von der Verteilungsdichte $p(f)$ der Funktionswerte abhängt. Man erkennt sofort, dass sich eine *lineare Quantisierungskennlinie* – gekennzeichnet durch *äquidistante* b_ν und a_ν – nur für *gleichverteilte* Funktionswerte ergibt. In diesem Falle geht nämlich (2.1.40) über in

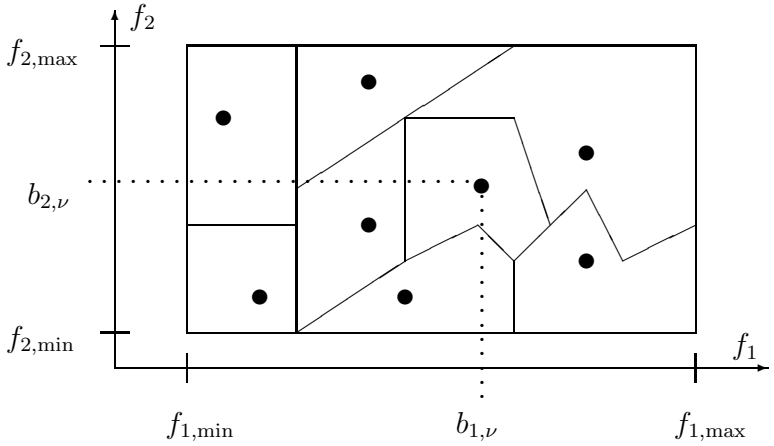
$$b_\nu = \frac{a_{\nu+1} + a_\nu}{2} \; . \tag{2.1.41}$$

Die Quantisierungsstufen haben dann die konstante Größe

$$a_{\nu+1} - a_\nu = \frac{a_{L+1} - a_1}{L} = \frac{f_{\max} - f_{\min}}{L} \; . \tag{2.1.42}$$

Bei nicht gleichförmiger Verteilung der Funktionswerte ergibt sich i. Allg. eine nichtlineare Quantisierungskennlinie. Aus (2.1.38) entnimmt man, dass die Stufen eng liegen sollten, wo häufig Funktionswerte auftreten, damit der Fehler ε klein bleibt. Eine nichtlineare Quantisierungskennlinie lässt sich einfach dadurch erreichen, dass man die Funktionswerte zunächst an einer nichtlinearen Kennlinie verzerrt und die verzerrten Funktionswerte dann linear quantisiert. Natürlich muss nun nach der Dekodierung eine entsprechende Entzerrung vorgenommen werden. In der Sprach– und Bildverarbeitung wird häufig eine logarithmische Verzerrung durchgeführt, d. h. man kodiert nicht $f(x, y)$ sondern $\log[f(x, y)]$.

Meistens ist eine PCM Darstellung Grundlage der digitalen Verarbeitung von Mustern, unter Umständen auch Ausgangspunkt einer anderen Art der Kodierung. Die PCM Darstellung erhält man nämlich relativ leicht durch geeignete Wandler, wie Mikrofon zur akustisch/elektrischen



Kodebuch B	
ν	$\boldsymbol{b}_\nu$
1	$(b_{1,1}, b_{2,1})$
$\vdots$	$\vdots$
ν	$(b_{1,\nu}, b_{2,\nu})$
$\vdots$	$\vdots$
L	$(b_{1,L}, b_{2,L})$

Bild 2.1.8: Ein durch L Vektoren, die im Kodebuch definiert sind, zu quantisierender Wertebereich in der Ebene.

oder Fernsehkamera zur optisch/elektrischen Wandlung, in Verbindung mit nichtlinearer Signalverstärkung zur Verzerrung, "sample–and–hold" Verstärkern zur Abtastung und Analog/Digital Wandlern zur Amplitudenquantisierung. Für verschiedene Zwecke werden dafür vollständige Geräte angeboten. Im Folgenden wird stets angenommen, dass $f_j \simeq f'_j$ ist; es wird also nicht mehr zwischen analogen Funktionswerten und ihrer quantisierten Darstellung unterschieden.

2.1.4 Vektorquantisierung

Im vorigen Abschnitt wurden skalare Größen, nämlich die Amplitudenwerte eines Sensorsignals, im Sinne des Fehlers ε in (2.1.38) optimal quantisiert. Das wesentliche Ergebnis fasst Satz 2.5 zusammen. Es ist eine naheliegende Verallgemeinerung, nicht nur jeweils *einen* Amplitudenwert zu quantisieren, sondern einen *Vektor*, der mehrere Werte enthält. Dieses führt auf die **Vektorquantisierung**, die inzwischen sowohl für die Kodierung von Signalen, insbesondere von Bildern, Bildfolgen und Sprache, als auch für die Klassifikation von Mustern, insbesondere in der *Worterkennung* und *Objekterkennung*, große Bedeutung erlangt hat. Sie lässt sich auch als ein Ansatz zum *unüberwachten Lernen* (s. Abschnitt 4.8) auffassen und ist ein Kandidat für die *Reduktion* einer sehr bzw. zu großen Stichprobe auf eine kleinere Zahl von typischen Elementen, was z. B. bei nichtparametrischen Dichteschätzungen wie der PARZEN-Schätzung (s. Abschnitt 4.2.6) von Interesse sein kann.

Die Verallgemeinerung von Bild 2.1.7 auf die Quantisierung von zweidimensionalen Werten ist in Bild 2.1.8 gezeigt. Ein zweidimensionaler Wertebereich ist in der Ebene durch die Werte $(f_{1,\min}, f_{1,\max})$ und $(f_{2,\min}, f_{2,\max})$ definiert. Ein beobachteter Wert $\boldsymbol{f}$ aus diesem Intervall soll durch Zuweisung eines Prototypvektors bzw. durch ein *Kodewort* $\boldsymbol{b}_\nu \in \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L\}$ kodiert werden. Die zum vorigen Abschnitt analoge Frage ist nun, wie der Wertebereich in Intervalle V_ν zu zerlegen ist und welches Kodewort einem Intervall zugeordnet wird. Wenn die Kodewörter einmal in einem **Kodebuch** gespeichert werden, reicht zur Kodierung eines Vektors $\boldsymbol{f}$ offenbar die Angabe einer ganzen Zahl $\nu \in \{1, \ldots, L\}$ aus. Dieses Prinzip gilt für die Kodierung von n–dimensionalen Vektoren. Gesucht ist eine Zerlegung des Wertebereichs in Intervalle V_ν und ein Kodebuch, das den Quantisierungsfehler minimiert.

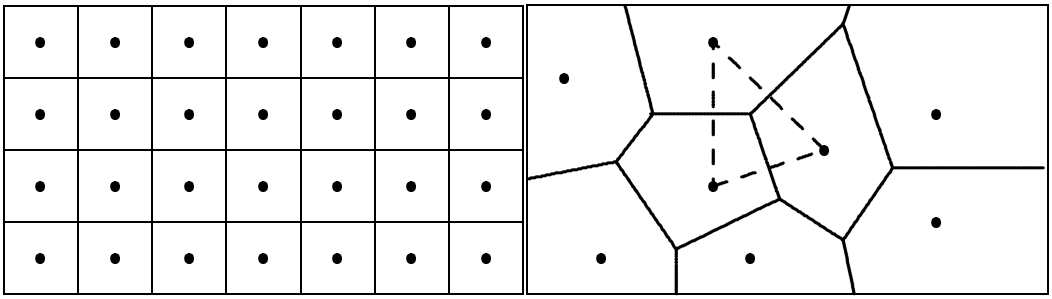


Bild 2.1.9: Beispiele für die Zerlegung einer Ebene und zugeordnete Kodewörter.

Der Quantisierungsfehler ist in direkter Verallgemeinerung von (2.1.38)

$$\varepsilon = \sum_{\nu=1}^{L} \int_{V_\nu} |\boldsymbol{f} - \boldsymbol{b}_\nu|^2 p(\boldsymbol{f}) \, \mathrm{d}\boldsymbol{f} \; . \tag{2.1.43}$$

Die Intervallgrenzen waren in (2.1.38) und Bild 2.1.7 Punkte auf der reellen Achse, während es nun im Prinzip beliebige Linien in der Ebene bzw. im n–dimensionalen Fall beliebige Teilvolumina V_ν des R^n sind. Daher ist die geschlossene Lösung durch Ableiten nach den Intervallgrenzen und den Kodewörtern hier nicht möglich. Die Teilvolumina und Kodewörter lassen sich jedoch iterativ bestimmen, wofür der folgende Satz die Basis bildet.

Satz 2.6 *Wenn ein Kodebuch $B = \{\boldsymbol{b}_\nu \mid \nu = 1, \ldots, L\}$ gegeben ist, dann erfordert die Minimierung des Fehlers ε, dass Vektoren $\boldsymbol{f}$ den Volumina V_ν zugeordnet werden gemäß der Regel*

$$\text{wenn } |\boldsymbol{f} - \boldsymbol{b}_\kappa|^2 \leq |\boldsymbol{f} - \boldsymbol{b}_\nu|^2 \text{ für alle } \nu \neq \kappa \; , \quad \text{dann} \quad \boldsymbol{f} \in V_\kappa \; . \tag{2.1.44}$$

Wenn eine Zerlegung des R^n in Teilvolumina V_ν gegeben ist, dann sind die besten Kodewörter definiert durch

$$\boldsymbol{b}_\nu = E\{\boldsymbol{f} \mid \boldsymbol{f} \in V_\nu\} \; , \quad \nu = 1, \ldots, L \; . \tag{2.1.45}$$

Die Regel (2.1.44) wird in Abschnitt 4.2.5 als *Minimumabstandsklassifikator* eingeführt. Sie ist intuitiv einleuchtend, da ein beobachteter Wert $\boldsymbol{f}$ dem Volumen V_ν zugeordnet wird, zu dessen Kodewort es den kleinsten Abstand hat. Der beste Prototypvektor bzw. das beste Kodewort für ein Teilvolumen V_ν ist nach (2.1.45) der Mittelwert der in diesem Volumen beobachteten Werte. Er wird numerisch geschätzt aus der Gleichung

$$\boldsymbol{b}_\nu = \frac{1}{N_\nu} \sum_{\boldsymbol{f} \in V_\nu} \boldsymbol{f} \; , \tag{2.1.46}$$

wobei N_ν die Zahl der in V_ν beobachteten Werte $\boldsymbol{f}$ ist.

Aus (2.1.44) folgt, dass die Grenze zwischen zwei Intervallen durch die Mittelsenkrechte zwischen den Kodewörtern definiert ist. Daher ist die Zerlegung in Bild 2.1.8 nicht optimal im Sinne des Fehlers ε in (2.1.43). Bild 2.1.9 zeigt zwei Beispiele für Zerlegungen, die den Bedingungen von Satz 2.6 genügen.

Es bleibt nun noch zu klären, wie ein Iterationsverfahren zur Bestimmung des Kodebuches und damit der Kodewörter aussieht. Im Prinzip kommen dafür Algorithmen zur Ermittlung

gegeben: Trainingsmenge $\omega = \{{}^{\varrho}\boldsymbol{f} | \varrho = 1, \ldots, N\}$ von Werten;
initiales Kodebuch $B^{(0)} = \{\boldsymbol{b}_\nu | \nu = 1, \ldots, L\}$;
Schwellwert Θ für den Quantisierungsfehler

setze Iterationsschritt $m = 0$;
berechne Quantisierungsfehler $\varepsilon^{(0)}$ mit $B^{(0)}$ und ω

- setze Iterationsschritt $m = m + 1$
- berechne Zuordnung der Werte ${}^{\varrho}\boldsymbol{f} \in \omega$ mit aktuellem Kodebuch $B^{(m-1)}$
- berechne neue Kodewörter $\boldsymbol{b}_\nu, \nu = 1, \ldots, L$
- berechne neuen Fehler $\varepsilon^{(m)}$

UNTIL relativer Fehler $\frac{|\varepsilon^{(m-1)} - \varepsilon^{(m)}|}{\varepsilon^{(m)}} \leq \Theta$

Bild 2.1.10: Das Prinzip des LBG–Algorithmus.

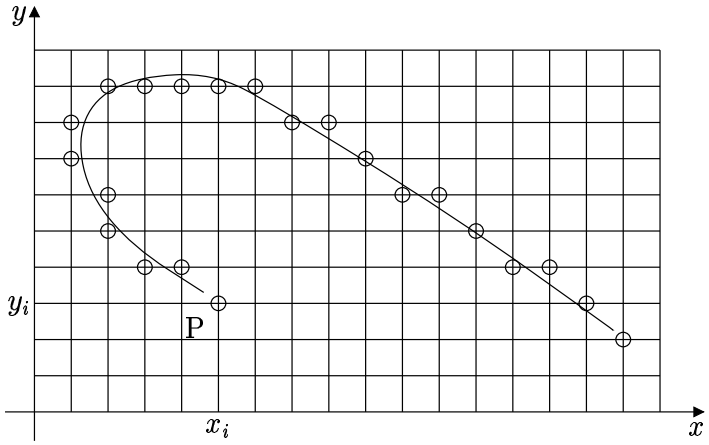
von Häufungsgebieten in Frage (s. Abschnitt 4.8.4). Ein Standardalgorithmus ist der **LBG–Algorithmus** (benannt nach den Autoren LINDE, BUZO, GRAY). Er basiert auf einer Iteration der Schritte Zuordnung von beobachteten Werten mit (2.1.44) und Neuschätzung von Kodewörtern mit (2.1.45). Unterschiede bestehen insbesondere in der Initialisierung des Kodebuches. Das Prinzip zeigt Bild 2.1.10. Die Iteration von Klassifikation und Neuschätzung ist auch Basis des *entscheidungsüberwachten Lernens* und der ISODATA–Algorithmen zur Ermittlung von Häufungsgebieten. Die Initialisierung des Kodebuches kann z. B. durch eine Menge zufällig erzeugter Kodewörter erfolgen, oder durch die sukzessive Aufspaltung von anfänglich zwei Kodewörtern. Diese werden gewählt zu

$$\boldsymbol{b}_{1,2} = (1 \pm \delta)\boldsymbol{\mu} \;, \quad \delta \ll 1 \;, \tag{2.1.47}$$

wobei $\boldsymbol{\mu}$ der Mittelwert der Stichprobe ω ist. Um die Gefahr des Hängenbleibens in lokalen Minima zu reduzieren, empfiehlt es sich, mehrere Initialisierungen zu verwenden, den LBG–Algorithmus darauf anzuwenden, und unter den Ergebnissen das beste auszuwählen.

2.1.5 Kodierung der Lauflänge

Eine spezielle Klasse von Bildern sind Schwarz-Weiß Bilder oder **binäre Muster**, bei denen nur die zwei Grauwerte Schwarz oder Weiß auftreten. Jeder Bildpunkt lässt sich also mit 1 bit kodieren, sodass für ein Bild $M_x \times M_y$ bit benötigt werden. In einem Bild werden in einer Bildzeile meistens mehrere aufeinander folgende Bildpunkte den gleichen Grauwert haben. In solchen Fällen lässt sich das Bild durch die sog. Lauflängen–Kodierung noch kompakter darstellen. Das Prinzip besteht darin, in einer Zeile nicht Punkt für Punkt die Grauwerte anzugeben, sondern die Zeile darzustellen durch Wertepaare (b_ν, l_k), wobei b_ν den Grauwert angibt und l_k die Lauflänge, d. h. die Zahl der aufeinander folgenden Punkte mit dem Grauwert b_ν. Grundsätzlich lässt sich also diese Kodierung auch auf Bilder mit mehr als zwei Grauwerten anwenden, jedoch wird bei mehr Grauwerten i. Allg. die Lauflänge kürzer und die Zahl der Paare (b_ν, l_k) je Zeile größer werden, sodass die Darstellung weniger effektiv ist. Das Problem der geeigneten Zuordnung von Kodeworten zu Paaren wird in der zitierten Literatur behandelt. Eine Alternative ist die Angabe des Grauwertsprungs (schwarz–weiß oder weiß–schwarz) und der Koordinaten seines Auftretens.



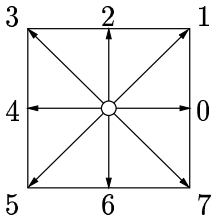


Bild 2.1.11: Kettenkodierung eines Linienmusters. Der Kode für die gezeigte Linie ist $P(x_i, y_j)$ 343232100007077077077

2.1.6 Kettenkodierung

Eine spezielle Klasse von Schwarz–Weiß Bildern sind Linienbilder, die nur dünne, schwarze Linien auf weißem Untergrund oder umgekehrt enthalten. Solche Muster lassen sich mit der in Bild 2.1.11 gezeigten **Kettenkodierung** darstellen. Von einem Startpunkt P beginnend wird jeweils die Richtung zum nächsten Punkt der Linie angegeben. Dabei werden nur acht Richtungen unterschieden, sodass 3 bit zur Kodierung einer Richtung ausreichen. Diese acht Richtungen ergeben die in (2.6.3), S. 136, definierte 8–Nachbarschaft. Je nach Art der Linienmuster kann es erforderlich sein, die Zahl der bit zu erhöhen, um z. B. auch Verzweigungen von Linien darzustellen. Einige Parameter der Linie, wie Linienlänge und Fläche zwischen Linie und x–Achse lassen sich direkt aus dem Kettenkode berechnen. Für einige regelmäßige Linienmuster kann man den Kettenkode aus der erzeugenden Funktion ableiten und auch einige Transformationen, wie Vergrößerung um den Faktor s, über diese ausführen.

Aus Bild 2.1.11 geht hervor, dass die Darstellung von Linien in einem Raster i. Allg. nur näherungsweise möglich ist, wobei die erreichte Genauigkeit von der Feinheit des Rasters abhängt. Ebenso ist in dem gezeigten rechteckigen Raster eine Drehung des Musters i. Allg. mit Verzerrungen verbunden.

Teilweise wird der Kettenkode in der Weise vereinfacht, dass nur die vier Richtungen 0,2,4 und 6 in Bild 2.1.11 verwendet werden, für deren Kodierung zwei bit ausreichen. Der Quantisierungsfehler wird damit natürlich größer. Außer Linienbildern lassen sich mit dem Kettenkode beispielsweise auch die Umrisslinien von Objekten kodieren.

2.1.7 Ergänzende Bemerkungen

Das Gebiet der Kodierung wurde hier nur kurz behandelt. Dabei wurden die für die digitale Verarbeitung von Mustern grundlegenden Verfahren berücksichtigt, dagegen fast alle Verfeinerungen, die eine Reduzierung der erforderlichen Zahl der bit – z. B. durch Verwendung optimaler Kodes – zum Ziele haben, ausgelassen. Das für die Speicherung und Übertragung von Signalen wichtige Gebiet der fehlererkennenden und fehlerkorrigierenden Kodes wurde ausgelassen, da diese für die digitale Verarbeitung von Mustern nicht von unmittelbarer Bedeutung sind.

Bild 2.2.1: Original und, von links nach rechts, Bild der Grauwerte, die größer oder gleich dem Schwellwert $\theta = 150$ sind (dieser Wert wurde so gewählt, das rund 50% der Grauwerte über dem Schwellwert liegen), sowie das Binärbild gemäß (2.2.1)

2.2 Schwellwertoperationen (VA.1.1.2, 27.12.2003)

2.2.1 Vorbemerkung

Bei Bildern von einfachen Objekten, die sich vor einem relativ homogenen Hintergrund befinden, ist vielfach die Verwendung von Grauwerten nicht erforderlich, da die interessierende Information auch aus einer Schwarz–Weiß Darstellung (oder binären Darstellung) hervorgeht. Ein Standardbeispiel sind die in Bild 1.2.2, S. 17, gezeigten Schriftzeichen, bei denen Grauwerte, zumindest für den Betrachter, auch nur störend wären; natürlich schließt das nicht aus, dass bei der Aufnahme zunächst Grauwerte gemessen werden. Aber auch bei anderen Mustern, wie z. B. Schaltzeichen oder Werkstücken, wird vielfach nur eine binäre Darstellung verwendet. Neben der Reduzierung des Speicheraufwandes wird auch die Verarbeitung der Muster vereinfacht.

Übliche Aufnahmegeräte, wie Fernsehkameras oder Zeilen von Fotodioden, liefern ein Signal, das monoton von der Bildhelligkeit abhängt, also eine Folge von Grauwerten. Eine binäre Darstellung sollte so gewählt sein, dass (fast) alle zum Objekt gehörenden Bildpunkte den Wert 1 (entsprechend „Weiß“) erhalten und (fast) alle zum Hintergrund gehörenden den Wert 0 (entsprechend „Schwarz“), oder auch umgekehrt. Wenn der Kontrast des Objekts gut ist, d. h. wenn sich die Grauwerte von Objekt und Hintergrund genügend unterscheiden, lässt sich die Binärisierung oder die Trennung von Objekt und Hintergrund im Prinzip durch die **Schwellwertoperation**

$$h_{jk} = \begin{cases} 1 & : \quad f_{jk} \geq \theta \\ 0 & : \quad \text{sonst} \end{cases} \tag{2.2.1}$$

durchführen, die aus einer Bildmatrix $\boldsymbol{f} = [f_{j,k}]$ mit z. B. $f_{j,k} \in \{0, 1, \ldots, 255\}$ eine Bildmatrix $\boldsymbol{h}$ mit $h_{jk} \in \{0, 1\}$ ergibt. Einen Eindruck von der Wirkung der Operation auf ein Bild, das *nicht* von dem Typ „relativ einfaches Objekt vor relativ homogenem Hintergrund“ ist, gibt Bild 2.2.1.

Allerdings führt ein fester oder *globaler* **Schwellwert** θ oft nicht zu befriedigenden Ergebnissen, da Inhomogenitäten in der Beleuchtung und den Reflektionseigenschaften von Objekt und Hintergrund sowie Rauschen des Aufnahmegerätes erhebliche Schwankungen der gemessenen Helligkeit zur Folge haben. In so einem Falle ist die Verwendung von mehreren *lokalen* Schwellwerten zweckmäßig. Dabei wird das Bild in *Blöcke* B_{mn} von $2^l \times 2^l$ Bildpunkten Größe zerlegt und je Block B_{mn} ein Schwellwert θ_{mn} berechnet; die Blockmitte liegt an der Bildposition (m, n).

Im Folgenden werden einige Verfahren zur automatischen Bestimmung eines Schwellwertes vorgestellt. Nach einer Schwellwertoperation, mit der Objektpunkte den Wert Eins und Hintergrundpunkte den Wert Null erhalten, lässt sich die *Konturlinie* eines Objekts relativ einfach bestimmen (s. Abschnitt 3.10.2). Hier werden nur Operationen nach (2.2.1) im *Ortsbereich* vorgestellt. Indem man ein (zweidimensionales) Bild $f(x, y)$ in ein dreidimensionales, binäres Volumen $V(x, y, z)$ überführt, lassen sich auch Schwellwertoperationen im Frequenzbereich ausführen. Das Volumen ist mit einer weiteren Koordinate z dadurch definiert, dass es den Wert $V(x, y, z) = 1$ bekommt für $0 < z < f(x, y)$ und $V = 0$ sonst. Für Einzelheiten dazu wird auf die Literatur verwiesen.

2.2.2 Grauwerthistogramm

Generell kann man die (kontinuierliche) Verteilungsdichte $p(f)$ der Werte einer Zufallsvariablen, z. B. des Grauwertes f eines Bildes $f(x, y)$, durch ein **Histogramm** mit diskreten Werten approximieren. Der Wertebereich $f_{\min} \leq f \leq f_{\max}$ wird in L in der Regel gleich große Intervalle $[a_1 = f_{\min}, a_2]$, $[a_2, a_3], \ldots, [a_L, a_{L+1} = f_{\max}]$ der Länge $(f_{\max} - f_{\min})/L$ eingeteilt. Die Wahrscheinlichkeit, einen Wert f im Intervall $[a_l, a_{l+1}]$ zu finden, kann bei bekanntem $p(f)$ berechnet werden; ein Schätzwert kann bestimmt werden, indem man für ein gegebenes Bild abzählt, wie oft Werte von f in diesem Intervall liegen. Das Intervall $[a_l, a_{l+1}]$ wird im Folgenden durch einen Index $l = 1, \ldots, L$ bezeichnet. Der durch Abzählen bestimmte Schätzwert der Wahrscheinlichkeit wird mit $\mathcal{H}(l\,|\,f)$ bezeichnet. Es gilt

$$\begin{aligned} P(f \in [a_l, a_{l+1}]) &= \int_{a_l}^{a_{l+1}} p(f)\,\mathrm{d}f\ , \\ \mathcal{H}(l\,|\,f) &= \frac{\text{Anzahl der Werte von } f \text{ im Intervall } [a_l, a_{l+1}]}{\text{Gesamtzahl der Werte von } f} \\ &= \widehat{P}(f \in [a_l, a_{l+1}]) \approx P(f \in [a_l, a_{l+1}])\ . \end{aligned} \tag{2.2.2}$$

Das gesamte Histogramm wird auch abgekürzt mit $\mathcal{H}$ bezeichnet und entspricht also $\mathcal{H}(l\,|\,f)$, $l = 1, \ldots, L$. Wenn ein Farbbild vorliegt, hat man die drei Komponenten $f_\mathrm{r}, f_\mathrm{g}, f_\mathrm{b}$, für die drei Farbkanäle. Für jeden Farbkanal kann ein einzelnes Histogramm $\mathcal{H}(l\,|\,f_\mathrm{c})$, $c \in \{\mathrm{r}, \mathrm{g}, \mathrm{b}\}$, berechnet werden oder auch ein dreidimensionales Histogramm $\mathcal{H}(l_1, l_2, l_3\,|\,f_\mathrm{r}, f_\mathrm{g}, f_\mathrm{b}) = \mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{f})$, $l_1 = 1, \ldots, L_1$, $l_2 = 1, \ldots, L_2$, $l_3 = 1, \ldots, L_3$. Die Histogramme $\mathcal{H}(l\,|\,f)$ bzw. $\mathcal{H}(l_1, l_2, l_3\,|\,f_\mathrm{r}, f_\mathrm{g}, f_\mathrm{b})$ werden auch als *Grauwerthistogramm* bzw. *Farbhistogramm* bezeichnet. Das Histogramm nach (2.2.2) enthält die relativen Häufigkeiten von Grauwerten, da es mit der Gesamtzahl der Werte normiert ist; es wird daher auch als normiertes Histogramm bezeichnet. Bei einem Muster $f(x, y)$ werden, wie in Abschnitt 2.1.3 erläutert, die Amplitudenwerte quantisiert, wie auch Bild 2.1.7, S. 72, zeigt, wobei auch dort L Intervalle verwendet wurden. Durch die Quantisierung werden alle Werte im Intervall $[a_l, a_{l+1}]$ durch einen Wert b_l approximiert. Im Allgemeinen kann die Zahl der Intervalle des Grauwerthistogramms verschieden von der Zahl der Amplitudenstufen sein, allerdings wird vielfach die gleiche Zahl verwendet.

Schwellwertverfahren arbeiten oft mit dem *Grauwerthistogramm* $\mathcal{H}(l\,|\,f)$. Eine Modifikation ist die Verwendung eines durch lokale Bildeigenschaften *gewichteten Grauwerthistogramms*. Die Werte des gewichteten Histogramms lassen sich nicht mehr sinnvoll als Wahrscheinlichkeiten interpretieren. Lokale Bildeigenschaften sind insbesondere die in Abschnitt 2.3.4 erwähnten Bildkontraste bzw. -konturen. Im Allgemeinen wird zur Ermittlung lokaler Bildeigenschaften ein gegebenes Bild $[f_{j,k}]$ in ein gemäß (2.3.1), S. 87, *vorverarbeitetes Bild* $[h_{j,k}] = T\{[f_{jk}]\}$

transformiert. Ein Beispiel für eine solche Transformation ist der Betrag des in (2.3.53), S. 103, eingeführten LAPLACE-*Operators*

$$h_{jk} = T\{[f_{jk}]\} = |(f_{j,k-1} + f_{j,k+1} + f_{j+1,k} + f_{j-1,k})/4 - f_{jk}| \ . \tag{2.2.3}$$

Sie ergibt große Werte im Bereich von *Änderungen* des Grauwertes und kleine Werte im Bereich von *fast konstanten* Grauwerten. Die Gewichtung erfolgt durch eine geeignet gewählte Funktion $\phi\,(h_{j,k})$ der lokalen Eigenschaften. Beispiele für Gewichtungsfunktionen ϕ sind

$$\phi'_{j,k} = h_{j,k} \ , \tag{2.2.4}$$

$$\phi''_{j,k} = \frac{1}{1 + h_{j,k}} \ . \tag{2.2.5}$$

Bei Verwendung der Gewichtungsfunktion (2.2.4) liefern Bildpunkte im Bereich von Änderungen im Grauwert *große* Beiträge im Histogramm, Bildpunkte im Bereich relativ homogener Grauwerte *kleine* Beiträge. Im Histogramm ergeben sich also große Werte im Bereich von Grauwertänderungen. Bei Verwendung der Gewichtungsfunktion (2.2.5) liefern Bildpunkte im Bereich von Änderungen im Grauwert *kleine* Beiträge im Histogramm, Bildpunkte im Bereich relativ homogener Grauwerte *große* Beiträge; relative Maxima und Minima im gewichteten Histogramm werden dadurch i. Allg. ausgeprägter als im ungewichteten.

Definition 2.3 *a) Das (normierte)* **Grauwerthistogramm** $\mathcal{H}(l\,|\,f)$ *eines Bildes* $[f_{jk}]$ *ist definiert durch*

$$\mathcal{H}(l\,|\,f) = \frac{M(b_l)}{M_x M_y} = \frac{\sum\limits_{f_{jk}=b_l} 1}{M} \ , \quad l = 1, \ldots, L \ , \tag{2.2.6}$$

wobei $M(b_l)$ *die Zahl der Bildpunkte* f_{jk} *mit Grauwert* $f_{jk} = b_l$ *ist,* $M = M_x \times M_y$ *die Gesamtzahl der Bildpunkte und* M_x *bzw.* M_y *die Zahl der Bildpunkte in* x*– bzw.* y*–Richtung.*

b) Das **gewichtete Grauwerthistogramm** *wird mit* $\mathcal{H}(l\,|\,f, \phi, T)$ *bezeichnet und ist gegeben durch*

$$\mathcal{H}(l\,|\,f, \phi, T) = \frac{\sum\limits_{f_{jk}=b_l} \phi\,(T\{[f_{jk}]\})}{M} \ , \quad l = 1, \ldots, L \ . \tag{2.2.7}$$

Es kann nützlich sein, das Histogramm zu *glätten*. Dieses ist möglich durch ein *Tiefpassfilter* (s. Abschnitt 2.3), durch eine Reduktion der anfänglichen Zahl L der Grauwertstufen vor Berechnung des Histogramms und durch Berechnung der Zahl $M(b_l)$ in (2.2.6) nicht für einen Grauwert b_l sondern über l_0 benachbarte Grauwerte, sodass die Zahl $M([b_l, \ldots, b_{l+l_0-1}])$ verwendet wird. Eine Vorverarbeitung des Bildes, z. B. zum Zwecke der Reduktion von Störungen, ist ebenfalls nützlich. Operationen dafür werden in Abschnitt 2.3 und Abschnitt 2.3.5 angegeben. Bild 2.2.2 zeigt einige Beispiele.

Es gibt zahlreiche weitere Modifikationen zur Berechnung des Grauwerthistogramms. Eine ist die Verwendung des sog. **gefilterten Histogramms**. Dieses wird nur unter Verwendung der Grauwerte von Punkten, in denen der Wert des LAPLACE-Operators einen Schwellwert übersteigt, berechnet. Der LAPLACE-Operator nimmt etwa gleich große Beträge zu beiden Seiten einer Grauwertkante an und ist sonst fast Null. Damit werden im Histogramm etwa gleich viele Punkte mit großem und kleinem Grauwert berücksichtigt, sodass die beiden Maxima etwa

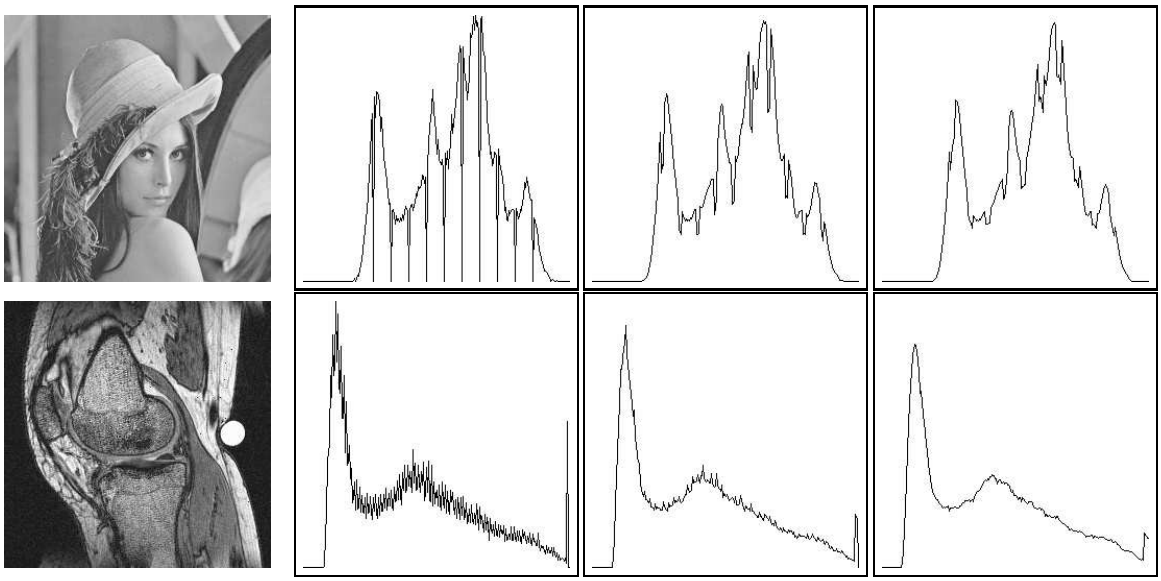


Bild 2.2.2: Jeweils von links nach rechts ist gezeigt: ein Grauwertbild; das Histogramm berechnet je Grauwertstufe; das Histogramm berechnet durch Zusammenfassen von $l_0 = 2$ Grauwertstufen; das Histogramm berechnet durch Zusammenfassen von $l_0 = 4$ Grauwertstufen

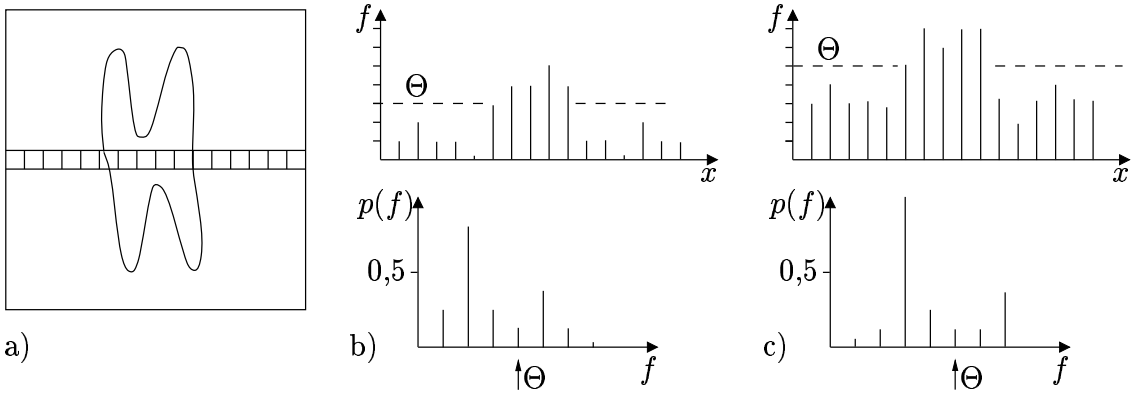


Bild 2.2.3: a) Ein Objekt, angedeutet durch seine Kontur, vor einem Hintergrund und eine Zeile von Abtastpunkten; b) gemessene Grauwerte längs der Zeile und relative Häufigkeiten $p(f)$ der Grauwerte; c) dasselbe wie in b), jedoch bei anderer Beleuchtung

gleich groß und daher besser trennbar sind. Eine Erweiterung ergibt sich durch die Berechnung eines zweidimensionalen Histogramms aus relativer Häufigkeit von Grauwerten und von Änderungen der Grauwerte, d. h. des *Grauwertgradienten*. Die Hinzunahme von Farbe führt zu dem oben bereits erwähnten *Farbhistogramm*.

2.2.3 Schwellwerte aus dem Grauwerthistogramm

Minimum im Grauwerthistogramm

In Bild 2.2.3 ist ein häufig verwendetes Verfahren zur Anpassung der Schwelle θ an das beobachtete Muster gezeigt. Man ermittelt für die gemessenen Grauwerte deren *Grauwerthistogramm*. Es ist zu erwarten, dass dieses bei einfachen Objekten mit einigermaßen homogener Oberfläche vor einem einigermaßen homogenen Hintergrund zwei relative Maxima hat, d. h. man erhält in etwa ein *bimodales Histogramm*. Durch die Verwendung eines gewichteten Histogramms wird, wie oben erwähnt, in solchen Fällen die Bimodalität i. Allg. ausgeprägter. Das eine Maximum wird von Bildpunkten des Objekts verursacht, das andere von denen des Hintergrunds.

Als Schwelle wählt man das relative Minimum zwischen diesen Maxima. Wenn die Lage des Minimums nicht eindeutig ist, wie in Bild 2.2.3c, so kann man z. B. den diskreten Grauwert wählen, der der mittleren Lage des Minimums am nächsten liegt und der dem Mittelwert zwischen den beiden Maxima am nächsten liegt. Modifikationen des Verfahrens ergeben sich insbesondere durch die Wahl des lokalen Bildausschnittes, in dem das Histogramm berechnet wird.

Hier ist die oben erwähnte Berechnung lokaler Schwellwerte in einer Blockzerlegung des Bildes nützlich, wobei die Blockgröße etwa im Bereich 11×11 Punkte liegen kann, d. h. es wird je ein Grauwerthistogramm und je ein Schwellwert pro Block berechnet. Vielfach werden Schwellwerte auch interaktiv festgelegt, indem man verschiedene Werte ausprobiert und das Ergebnis subjektiv beurteilt. Die Festlegung eines Schwellwertes, der für eine Menge von Bildern, bzw. eine Stichprobe ω von Bildern aus dem Problemkreis, gute Ergebnisse bringt, ist interaktiv natürlich mühsam.

Schnittpunkt zweier Normalverteilungen

Die oben erwähnte Bimodalität des Grauwerthistogramms legt es nahe, dieses durch die Addition zweier Normalverteilungen zu approximieren und als Schwellwert den Schnittpunkt der Normalverteilungen zu wählen, wie in Bild 2.2.4 angedeutet. Die Summe der Normalverteilungen ist

$$p(f) = \frac{\alpha}{\sigma_1\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{f-m_1}{\sigma_1}\right)^2\right] + \frac{1-\alpha}{\sigma_2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{f-m_2}{\sigma_2}\right)^2\right] \qquad (2.2.8)$$

mit dem Grauwert f und den unbekannten Parametern $\alpha, m_1, \sigma_1, m_2, \sigma_2$. Dieses ist ein einfaches Beispiel für eine *Mischung* zweier Normalverteilungen, die in Abschnitt 4.2.1 verallgemeinert betrachtet wird. Die exakte Lösung zur Berechnung der Parameter und des Schnittpunkts der Normalverteilungen ist aufwendig und wird daher durch eine Näherungslösung ersetzt. Man wählt einen Schwellwert θ_0, mit $f_{\min} \leq \theta_0 \leq f_{\max}$. Dann berechnet man m_1, σ_1 mit Werten *links* von θ_0 sowie m_2, σ_2 mit Werten *rechts* davon und α aus dem Verhältnis. Für diese Approximation berechnet man den Fehler zwischen $p(f)$ und dem Grauwerthistogramm und verändert den Schwellwert bis der Fehler minimal ist. Da die Grauwerte f auf L Stufen quantisiert sind, gibt es nur wenige diskrete Schwellwerte auszuprobieren.

Unimodales Grauwerthistogramm

Die obigen beiden Verfahren versagen natürlich, wenn das Histogramm (fast) unimodal ist, d. h. nur *ein* ausgeprägtes relatives Maximum aufweist. Eine einfache Operation zur Bestimmung

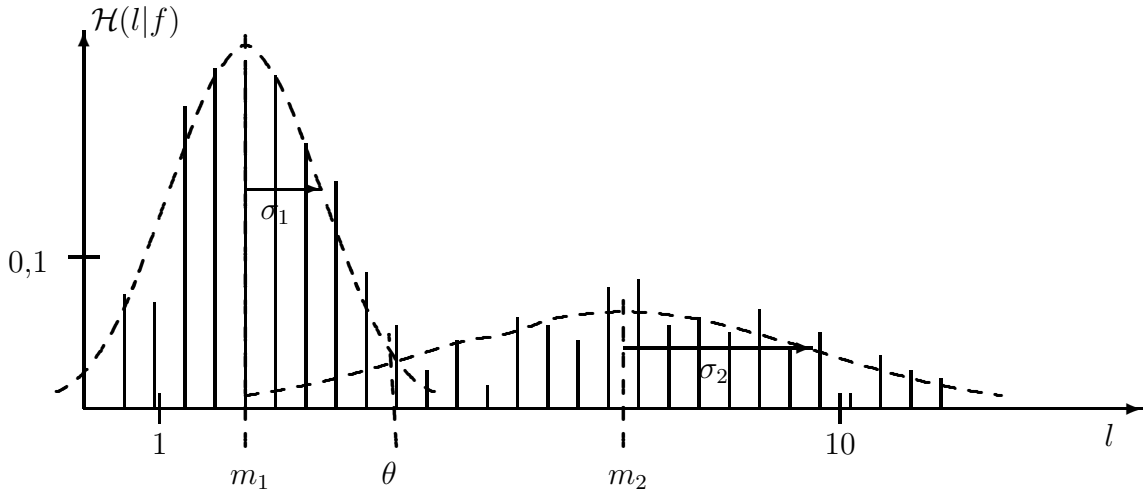


Bild 2.2.4: Ein Grauwerthistogramm wird durch die Summe zweier Normalverteilungen approximiert

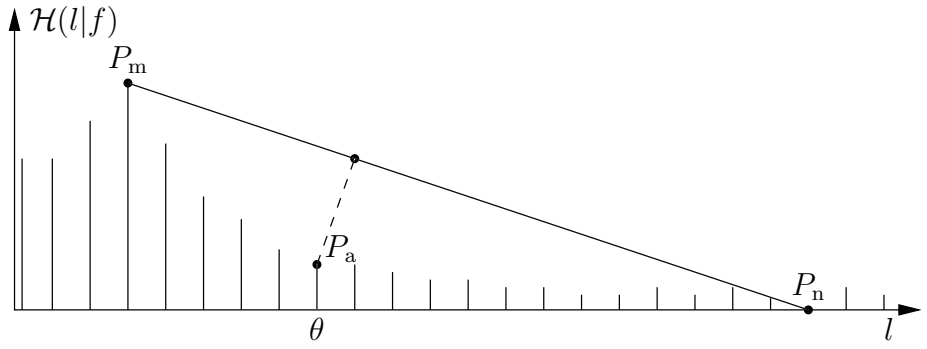


Bild 2.2.5: Der Schwellwert wird aus relativem Maximum und erstem verschwindenden Wert im Histogramm wie skizziert bestimmt

eines Schwellwertes in diesem Fall zeigt Bild 2.2.5. Es wird der Punkt P_m des relativen Maximums bestimmt sowie der Punkt P_n, an dem zum ersten Mal die Häufigkeit Null nach einer von Null verschiedenen Häufigkeit im Histogramm auftritt. Zwischen $P_\mathrm{m}, P_\mathrm{n}$ wird eine Gerade gelegt und der Punkt P_a im Histogramm bestimmt, der den größten senkrechten Abstand von der Geraden hat. Mit $P_\mathrm{a} = (f_\mathrm{P}, p(f_\mathrm{P}))$ ergibt sich der Schwellwert θ aus $\theta = f_\mathrm{P}$.

2.2.4 Optimierte Schwellwerte

Eine Optimierung der Schwellwertbestimmung kann wie folgt durchgeführt werden. Wie oben wird von L Grauwertstufen $b_1, \ldots, b_L$ ausgegangen. Die Wahrscheinlichkeit, dass ein Bildpunkt den Grauwert b_ν hat, sei

$$p(f = b_\nu) \;=\; p_\nu\,, \quad \nu = 1, \ldots, L \tag{2.2.9}$$

und wird mit dem Grauwerthistogramm geschätzt. Mit einem Schwellwert $\theta = b_l$ wird die Menge der Bildpunkte durch (2.2.1) in zwei Klassen

$$\Omega_1^l = \{f_{jk} \,|\, f_{jk} \leq b_l\} \quad \text{bzw.} \quad \Omega_2^l = \{f_{jk} \,|\, f_{jk} > b_l\} \tag{2.2.10}$$

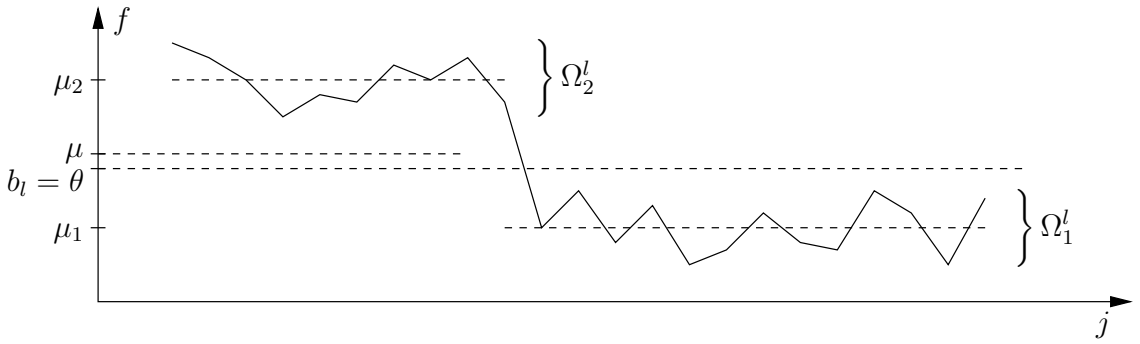


Bild 2.2.6: Die Grauwerte in einem Bild werden durch nur zwei Werte $\{\mu_1, \mu_2\}$, die durch die Werte $\{0, 1\}$ kodiert werden, approximiert

zerlegt. Die Wahrscheinlichkeit, dass ein Punkt zu Ω_1^l bzw. Ω_2^l gehört, ist

$$p(\Omega_1^l) = \sum_{\nu=1}^{l} p_\nu \quad \text{bzw.} \qquad p(\Omega_2^l) = \sum_{\nu=l+1}^{L} p_\nu \;=\; 1 - p(\Omega_1^l)\;. \tag{2.2.11}$$

Die bedingten mittleren Grauwerte μ_1 und μ_2 der Punkte in Ω_1^l und Ω_2^l sowie der mittlere Grauwert μ des Bildes sind

$$\mu_1 \;=\; \sum_{\nu=1}^{l} b_\nu p(f = b_\nu \,|\, \Omega_1^l) = \sum_{\nu=1}^{l} \frac{b_\nu p_\nu}{p(\Omega_1^l)}\;, \tag{2.2.12}$$

$$\mu_2 \;=\; \sum_{\nu=l+1}^{L} b_\nu p(f = b_\nu \,|\, \Omega_2^l) = \sum_{\nu=l+1}^{L} \frac{b_\nu p_\nu}{p(\Omega_2^l)}\;, \tag{2.2.13}$$

$$\mu \;=\; \sum_{\nu=1}^{L} b_\nu p_\nu \;=\; \mu_1\, p(\Omega_1^l) + \mu_2\, p(\Omega_2^l)\;. \tag{2.2.14}$$

Bild 2.2.6 zeigt die Verhältnisse am Beispiel einer Bildzeile.

Ein erstes sinnvolles Kriterium $G_l^{(1)}$ für die Güte der Klassen bzw. die Güte des Schwellwertes $\theta = b_l$ ergibt sich aus der Forderung, dass die Klassen möglichst viele Elemente enthalten sollten und dass ihre bedingten Mittelwerte möglichst verschieden sein sollten, zu

$$G_l^{(1)} \;=\; p(\Omega_1^l)p(\Omega_2^l)(\mu_2 - \mu_1)^2\;. \tag{2.2.15}$$

Wird die Schwelle $\theta = b_l$ zu weit gesenkt, so wird $p(\Omega_1^l) \simeq 0$, und bei zu hoher Schwelle wird $p(\Omega_2^l) \simeq 0$. In beiden Fällen ist $G_l^{(1)} \simeq 0$, und dazwischen liegt ein Maximum von $G_l^{(1)}$. Als Schwellwert wird der Wert $\theta = b_{l^*}$ bestimmt, für den $G_l^{(1)}$ maximiert wird, also

$$G_{l^*}^{(1)} = \max_{l \in \{1,\ldots,L\}} G_l^{(1)} \quad \Longrightarrow \quad \theta = b_{l^*} = \operatorname*{argmax}_{l \in \{1,\ldots,L\}} G_l^{(1)}\;. \tag{2.2.16}$$

Zur effizienten Berechnung führt man noch eine Größe

$$\mu(l) = \sum_{\nu=1}^{l} b_\nu p_\nu = \mu_1\, p(\Omega_1^l) \tag{2.2.17}$$

ein. Berücksichtigt man $\mu = \mu_1\, p(\Omega_1^l) + \mu_2\, p(\Omega_2^l)$, so geht (2.2.15) über in

$$\boxed{G_l^{(1)} \;=\; \frac{\big(p(\Omega_1^l)\mu - \mu(l)\big)^2}{p(\Omega_1^l)\,\big(1 - p(\Omega_1^l)\big)}\;.} \tag{2.2.18}$$

Damit lässt sich $G_l^{(1)}$ für $l = 1, \ldots, L$ einfach berechnen und der Wert $l = l^*$ bestimmen, für den $G_l^{(1)}$ maximiert wird. Es genügt, $G_l^{(1)}$ für die Werte von l zu berechnen, für die $p(\Omega_1^l)p(\Omega_2^l) > 0$ ist. Man kann zeigen, dass die Wahl von θ gemäß (2.2.16) auch in dem Sinne optimal ist, dass man das Muster mit zwei Grauwerten mit minimalem mittleren quadratischen Fehler approximiert.

Wenn man mehrere verschiedene Objekte oder Objektteile mit unterschiedlichen Grauwerten vom Hintergrund trennen will, so ist die Einführung mehrerer Schwellwerte eine naheliegende Verallgemeinerung von (2.2.1). Das Ergebnis ist dann eine Bildmatrix $\boldsymbol{h}$ mit mehr als zwei Grauwerten. Die Quantisierung erfolgt gemäß

$$\begin{aligned} h_{jk} \;=\;& \nu - 1 \quad \text{wenn} \\ & b_{l(\nu-1)} < f_{jk} \le b_{l(\nu)} \quad , \quad \nu = 1, 2, \ldots, m \quad ; \\ & b_{l(0)} = b_1 - 1 \qquad\quad ; \quad b_{l(m)} = b_L \qquad\quad . \end{aligned} \tag{2.2.19}$$

Mit $m = 2$ und $\theta = b_{l(1)}$ geht (2.2.19) offensichtlich in (2.2.1) über. Man kann auch hier versuchen, die $b_{l(j)}$, $j = 1, \ldots, m-1$ aus den Minima des Grauwerthistogramms zu bestimmen, vorausgesetzt es gibt $m - 1$ genügend ausgeprägte Minima. Im Prinzip lässt sich auch die Vorgehensweise von (2.2.9) – (2.2.16) anwenden, da sich (2.2.10) – (2.2.14) sofort auf mehr als zwei Klassen verallgemeinern lassen. Aus (2.2.15) erhält man nämlich mit (2.2.14)

$$G_l^{(1)} = p(\Omega_1^l)\,(\mu_1 - \mu)^2 + p(\Omega_2^l)\,(\mu_2 - \mu)^2 \;. \tag{2.2.20}$$

Eine direkte Verallgemeinerung auf mehrere Schwellwerte ist dann

$$\boxed{G^{(1)}_{l(1),\ldots,l(m-1)} = \sum_{j=1}^{m} p(\Omega_j^{l(j)})\,(\mu_j - \mu)^2 \;,} \tag{2.2.21}$$

und die Schwellwerte ergeben sich analog zu (2.2.16) aus den Werten $l^*(1), \ldots, l^*(m-1)$ für die G in (2.2.21) maximiert wird. Allerdings wird die erforderliche Suche mit wachsendem m immer aufwendiger, sodass das Verfahren auf $m = 2$ bis $m = 4$ beschränkt sein dürfte. In der Regel werden ohnehin nur ein oder zwei Schwellwerte verwendet, d. h. $m \le 3$ in (2.2.19).

Eine Erweiterung des obigen Ansatzes zur Berechnung optimierter Schwellwerte ist die Hinzunahme der *Streuungen* σ_1, σ_2 der Grauwerte der beiden Klassen

$$\begin{aligned} \sigma_1^2 \;&=\; \sum_{\nu=1}^{l} (b_\nu - \mu_1)^2\; p(f = b_\nu | \Omega_1^l) \;, \\ \sigma_2^2 \;&=\; \sum_{\nu=l+1}^{L} (b_\nu - \mu_2)^2\; p(f = b_\nu | \Omega_2^l) \;. \end{aligned} \tag{2.2.22}$$

Die obige Gleichung (2.2.20), die sich in (2.2.15) bzw. in (2.2.18) umformen lässt, ist ein Maß für die Streuung σ_z *zwischen* den beiden Klassen, und

$$\sigma_i = p(\Omega_1^l)\sigma_1^2 + p(\Omega_2^l)\sigma_2^2 \tag{2.2.23}$$

ist ein Maß für die Streuung *innerhalb* dieser Klassen. Für eine gute Klassentrennung sollte σ_z *groß* und σ_i *klein* sein. Ein geeignetes Gütekriterium für den Schwellwert ist daher auch das Verhältnis dieser beiden Streuungen, also

$$\boxed{G_l^{(2)} = \frac{p(\Omega_1^l)\,(\mu_1 - \mu)^2 + p(\Omega_2^l)\,(\mu_2 - \mu)^2}{p(\Omega_1^l)\sigma_1^2 + p(\Omega_2^l)\sigma_2^2}\,.} \tag{2.2.24}$$

Dieses Kriterium ist aus der *Diskriminanzanalyse* (s. Abschnitt 3.8.2) bekannt. Auch hier wird analog zu (2.2.16) der Wert l^* bestimmt, für den $G_l^{(2)}$ *maximiert* wird. Die Verallgemeinerung von (2.2.24) auf mehrere Schwellwerte, analog zu (2.2.21), ist offensichtlich.

2.2.5 Unsicherheit und Homogenität

Ein aufwendiger zu berechnender Schwellwert kann aus der Kombination von informationstheoretisch gemessener Unsicherheit der beiden Klassen „Objekt“ und „Hintergrund“ mit der Homogenität der entstehenden Klassenregionen bestimmt werden.

Wie oben in Abschnitt 2.2.4 werden mit dem Schwellwert $\theta = b_l$ die zwei Klassen Ω_1^l bzw. Ω_2^l unterschieden, die die a priori Wahrscheinlichkeiten $p(\Omega_1^l)$ bzw. $p(\Omega_2^l)$ haben. Die *a posteriori Wahrscheinlichkeiten* der beiden Klassen sind

$$\begin{aligned} p(\Omega_1^l \,|\, f_{j,k} = b_\nu) &= \frac{p(f_{j,k} = b_\nu \,|\, \Omega_1^l)\, p(\Omega_1^l)}{p(f_{j,k} = b_\nu)} \\ p(\Omega_2^l \,|\, f_{j,k} = b_\nu) &= \frac{p(f_{j,k} = b_\nu \,|\, \Omega_2^l)\,(1 - p(\Omega_1^l))}{p(f_{j,k} = b_\nu)}\,. \end{aligned} \tag{2.2.25}$$

Die Wahrscheinlichkeiten $p(f_{j,k} = b_\nu \,|\, \Omega_\kappa^l)$, $\kappa = 1, 2$, können z. B. durch Normalverteilungen approximiert werden. Die *Entropie* der beiden a posteriori Wahrscheinlichkeiten ist ein Maß für die *Unsicherheit*, einen Bildpunkt nach Ω_1 bzw. Ω_2 zu klassifizieren, wenn man den Grauwert $f_{j,k} = b_\nu$ beobachtet hat und den Schwellwert $\theta = b_l$ verwendet. Die Klassenunsicherheit $H^l(f_{j,k})$ ist daher

$$H^l(f_{j,k}) = -\sum_{\kappa=1}^{2} p(\Omega_\kappa^l \,|\, f_{j,k} = b_\nu) \log\left[p(\Omega_\kappa^l \,|\, f_{j,k} = b_\nu)\right]\,. \tag{2.2.26}$$

Damit kann die Klassenunsicherheit für jeden Schwellwert und jeden Grauwert berechnet werden.

Die Homogenität einer Region von Grauwerten wird als Eigenschaft eines Bildpunktes $f_{j,k}$ definiert, die vom „Zusammenhang“ benachbarter Bildpunkte abhängt. Sie wird gemessen durch eine Funktion $h(f_{j,k})$, zu deren genauer Definition aus Platzgründen auf die Literatur verwiesen wird.

Damit wird eine *Schwellwertenergie* als kombiniertes Kriterium aus Klassenunsicherheit und Regionenhomogenität

$$\boxed{E(b_l) = \sum_{j,k} H^l(f_{j,k})h(f_{j,k}) + (1 - H^l(f_{j,k}))(1 - h(f_{j,k}))} \tag{2.2.27}$$

definiert, und der optimale Schwellwert ergibt sich aus

$$\theta = \operatorname{argmin} E(b_l) \ . \tag{2.2.28}$$

Diesem Kriterium liegt die Annahme bzw. Voraussetzung zu Grunde, dass Bildpunkte $f_{j,k}$ mit hoher Klassenunsicherheit gerade in der Nachbarschaft von Klassengrenzen, d. h. inhomogenen Bildbereichen, auftreten.

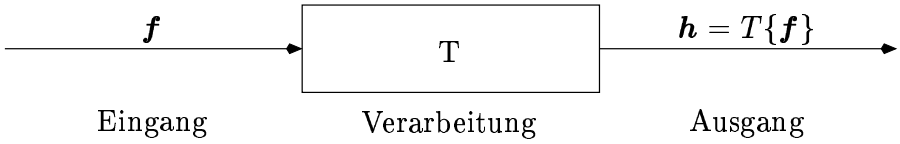


Bild 2.3.1: Vorverarbeitung eines Musters durch ein System, das eine Transformation T realisiert

2.3 Lineare Operationen (VA.1.4.2, 04.12.2005)

2.3.1 Anliegen

Muster können durch das Aufnahmeverfahren, die Übertragung oder auch bereits bei ihrer Entstehung in einer Weise beeinflusst werden, die für den menschlichen Betrachter störend ist. Beispiele sind die Zuordnung falscher Grau- oder Farbwerte zu einzelnen Bildpunkten oder die Überlagerung von Sprache mit einem Fremdgeräusch. Es ist naheliegend, die Reduzierung störender Einflüsse auf das Muster anzustreben, bzw. zu versuchen, ein möglichst „ideales“ Muster zu gewinnen. Die in (2.3.1) gezeigte grundsätzliche Vorgehensweise besteht darin, ein Muster $f(x, y)$ oder dessen PCM Darstellung $\boldsymbol{f} = [f_{jk}]$ mit einer geeigneten Transformation T in ein neues Muster

$$\boldsymbol{h} = [h_{jk}] = T\{[f_{jk}]\} \tag{2.3.1}$$

umzuwandeln, wobei T so gewählt wird, dass $\boldsymbol{h}$ für die weitere Verarbeitung besser geeignet ist als $\boldsymbol{f}$. Natürlich sind auch (2.2.1), (2.2.19), S. 84, spezielle Transformationen T, jedoch stand hier die Reduzierung von Störungen oder die Verbesserung der Qualität der Muster nicht im Vordergrund.

2.3.2 Lineare Systeme

Eine wichtige Klasse von Transformationen sind die **linearen Transformationen**, die durch ein **lineares System** realisiert werden.

Definition 2.4 *Wenn für zwei Funktionen ${}^1\boldsymbol{f}, {}^2\boldsymbol{f}$ und für zwei reelle Konstanten a_1, a_2 die Beziehung*

$$T\{a_1 {}^1\boldsymbol{f} + a_2 {}^2\boldsymbol{f}\} = a_1 T\{{}^1\boldsymbol{f}\} + a_2 T\{{}^2\boldsymbol{f}\} \tag{2.3.2}$$

gilt, heißt die Transformation T linear.

Da in diesem Buch nur die digitale Verarbeitung von Mustern erörtert wird, wird die folgende Diskussion auf die digitale Darstellung gemäß (2.1.1) beschränkt. Allerdings gelten entsprechende Beziehungen auch für den kontinuierlichen Fall, wie z. B. in (2.3.13) angegeben. Die Eigenschaften eines linearen Systems sind vollständig bekannt, wenn die Impulsantwort $\boldsymbol{g}$ des Systems bekannt ist.

Definition 2.5 *Definiert man einen (diskreten)* **Einheitsimpuls** *mit*

$$\delta_{jk} = \begin{cases} 1 & : \quad j = k = 0 \\ 0 & : \quad \text{sonst} \end{cases}, \tag{2.3.3}$$

so ist die **Impulsantwort** *definitionsgemäß die Reaktion des Systems auf einen Einheitsimpuls am Eingang. Es ist also*

$$g_{jk,\mu\nu} = T\{\delta_{j-\mu,k-\nu}\} \; . \tag{2.3.4}$$

In (2.3.4) kommt zum Ausdruck, dass die Impulsantwort i. Allg. davon abhängt, an welchem Ort (bzw. zu welcher Zeit) der Impuls aufgebracht wird. Die Bedeutung der Impulsantwort liegt darin, dass man mit ihr die Ausgangsgröße $\boldsymbol{h}$ für jede Eingangsgröße $\boldsymbol{f}$ berechnen kann. Das ist die Aussage von

Satz 2.7 *Für ein Eingangssignal $\boldsymbol{f} = [f_{jk}]$ ergibt sich das Ausgangssignal $\boldsymbol{h} = [h_{jk}]$ aus der Gleichung*

$$h_{jk} = \sum_{\mu=-\infty}^{\infty} \sum_{\nu=-\infty}^{\infty} f_{\mu\nu} g_{jk,\mu\nu} \; . \tag{2.3.5}$$

Beweis: Die Gleichung folgt unmittelbar aus (2.3.1) – (2.3.4). Man kann nämlich eine unendliche Folge von Abtastwerten

$$\boldsymbol{f} = [f_{jk} \,|\, j,k = 0, \pm 1, \pm 2, \ldots] \tag{2.3.6}$$

mit dem Einheitsimpuls auch als Summe

$$\boldsymbol{f} = \sum_{\mu=-\infty}^{\infty} \sum_{\nu=-\infty}^{\infty} f_{\mu\nu} \delta_{j-\mu,k-\nu} \; , \qquad j,k = 0, \pm 1, \ldots \tag{2.3.7}$$

schreiben. Für die Stelle (j,k) der Ausgangsgröße $\boldsymbol{h}$ gilt dann mit (2.3.1), (2.3.2)

$$\begin{aligned} h_{jk} &= T\{[f_{jk}]\} \\ &= \sum_{\mu} \sum_{\nu} f_{\mu\nu} T\{\delta_{j-\mu,k-\nu}\} \; , \end{aligned} \tag{2.3.8}$$

und daraus ergibt sich mit (2.3.4) sofort (2.3.5). Damit ist Satz 2.7 bewiesen.

Die Verwendung einer Impulsantwort gemäß (2.3.4) ist recht unhandlich, da die Speicherung von g für alle Indizes $(j,k;\mu,\nu)$ erforderlich ist. Bei der speziellen Klasse der **verschiebungsinvarianten Systeme** bewirkt jedoch eine Verschiebung des Einheitsimpulses lediglich eine entsprechende Verschiebung der Impulsantwort, es ist also

$$T\{\delta_{j-\mu,k-\nu}\} = g_{j-\mu,k-\nu} \; . \tag{2.3.9}$$

In diesem Fall kann man ohne Einschränkung der Allgemeinheit den Impuls stets an der Stelle $\mu = \nu = 0$ ansetzen und erhält die Impulsantwort des verschiebungsinvarianten Systems zu

$$\boxed{g_{jk} = T\{\delta_{jk}\} \; .} \tag{2.3.10}$$

Die Systemreaktion ergibt sich nun aus dem folgenden Satz.

Satz 2.8 *Wenn man auf den Eingang eines verschiebungsinvarianten linearen Systems mit der Impulsantwort* $[g_{jk}]$ *ein Muster* $[f_{jk}]$ *gibt, so erhält man die Ausgangsgröße* $[h_{jk}]$ *aus der* **diskreten Faltung** *von* $[f_{jk}]$ *und* $[g_{jk}]$ *zu*

$$\begin{aligned} h_{jk} &= \sum_{\mu=-\infty}^{\infty} \sum_{\nu=-\infty}^{\infty} f_{\mu\nu}\, g_{j-\mu,k-\nu} \\ &= \sum_{\mu=-\infty}^{\infty} \sum_{\nu=-\infty}^{\infty} f_{j-\mu,k-\nu}\, g_{\mu\nu} \qquad j,k = 0,\pm1,\pm2,\ldots \end{aligned} \tag{2.3.11}$$

Zur Abkürzung wird die Faltung auch symbolisch durch

$$\boldsymbol{h} = \boldsymbol{f} * \boldsymbol{g}\,, \quad \text{bzw.} \quad [h_{jk}] = [f_{jk}] * [g_{jk}] \tag{2.3.12}$$

dargestellt, wobei die Elemente h_{jk} der Folge $[h_{jk}]$ durch (2.3.11) gegeben sind.

Beweis: Der Beweis des ersten Teils von (2.3.11) folgt in offensichtlicher Weise aus dem Beweis von Satz 2.7, insbesondere aus (2.3.8). Der zweite Teil ergibt sich, wenn man $j-\mu = l, \quad k-\nu = m$ setzt.

Die Rechnungen wurden hier für Folgen von Abtastwerten durchgeführt. Für kontinuierliche Funktionen ergeben sich analoge Beziehungen. Insbesondere ist die Faltung einer kontinuierlichen Funktion $f(x,y)$ mit einer kontinuierlichen Gewichtsfunktion $g(x,y)$ gegeben durch das **Faltungsintegral**

$$h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)g(x-u,y-v)\,\mathrm{d}u\,\mathrm{d}v\,. \tag{2.3.13}$$

Dieses Integral steht in direkter Analogie zur *Faltungssumme* (2.3.11).

Für konkrete Rechnungen ist zu berücksichtigen, dass Muster i. Allg. nur in einem endlichen Intervall definiert sind oder auf ein solches mit genügender Genauigkeit beschränkt werden können; das gleiche gilt für die Impulsantwort. Damit reduzieren sich die unendlichen Summen in (2.3.11) auf endliche. Das Muster $f(x,y)$ werde wie in (2.1.1) mit M_xM_y Abtastwerten dargestellt, die Impulsantwort mit m_xm_y. Man kann sich vorstellen, dass außerhalb des in (2.1.2) gegebenen Bereiches $(x_0,x_1;y_0,y_1)$ das Muster identisch Null ist. Damit ergibt sich für (2.3.11)

$$\begin{aligned} h_{jk} &= \sum_{\mu=0}^{M_x-1} \sum_{\nu=0}^{M_y-1} f_{\mu\nu} g_{j-\mu,k-\nu} \\ &\quad j = 0,1,\ldots,M_x+m_x-2\,; \quad k = 0,1,\ldots,M_y+m_y-2\,, \end{aligned} \tag{2.3.14}$$

bzw. die äquivalente Form

$$\boxed{\begin{aligned} h_{jk} &= \sum_{\mu=0}^{m_x-1} \sum_{\nu=0}^{m_y-1} f_{j-\mu,k-\nu} g_{\mu\nu} \\ &\quad j = 0,1,\ldots,M_x+m_x-2\,, \quad k = 0,1,\ldots,M_y+m_y-2\,. \end{aligned}} \tag{2.3.15}$$

Da normalerweise $m_x < M_x$, $m_y < M_y$ sein wird, ist (2.3.15) vorzuziehen. Wenn man also zwei Funktionen $\boldsymbol{f}$ und $\boldsymbol{g}$ mit M_xM_y und m_xm_y Abtastwerten faltet, so hat das Ergebnis

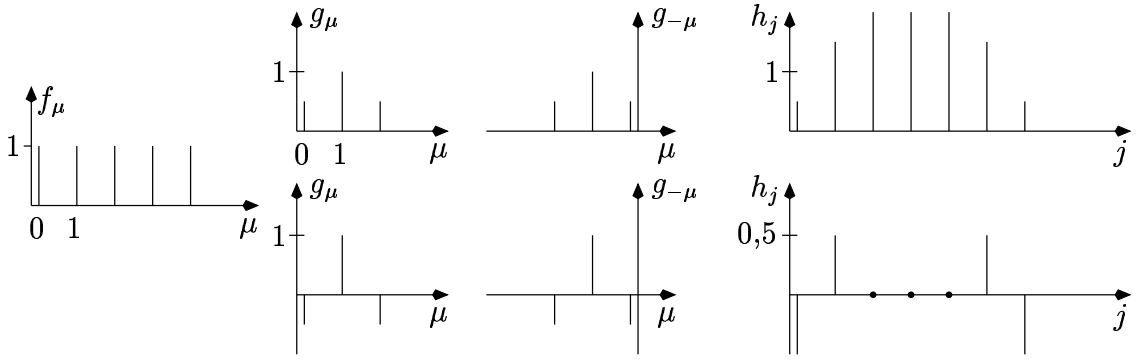


Bild 2.3.2: Beispiel für die Faltung einer Funktion $[f_\mu]$ mit zwei verschiedenen Impulsantworten $[g_\mu]$

$\boldsymbol{h}$ genau $(M_x + m_x - 1)(M_y + m_y - 1)$ Abtastwerte. Die in (2.3.14), (2.3.15) angegebenen Beziehungen lassen sich ohne weiteres auf Funktionen mit beliebiger Zahl von Variablen verallgemeinern. Die Faltung ist eine *lineare Nachbarschaftsoperation*.

Es ist bei Zeitfunktionen zu beachten, dass ein verschiebungsinvariantes System nur dann *kausal* ist, wenn

$$g_j = 0 \quad \text{für} \quad j < 0 \tag{2.3.16}$$

ist. Andernfalls würde wegen (2.3.10) die Systemreaktion bereits beginnen, ehe das Eingangssignal beginnt. Die Forderung nach **Kausalität** spielt nur bei Zeitfunktionen eine Rolle, aber nicht bei Ortsfunktionen, da die Ortskoordinaten in beiden Richtungen durchlaufen werden können. Es sei noch erwähnt, dass ein System, dessen Impulsantwort der Bedingung

$$\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} |g_{jk}| < \infty \tag{2.3.17}$$

genügt, als *stabil* bezeichnet wird. Ist $[f_{jk}]$ eine Funktion, deren Elemente $f_{jk} < A$ für irgendein endliches A und alle j, k sind, so heißt diese Funktion beschränkt. Ist eine beschränkte Funktion die Eingangsgröße eines **stabilen Systems**, dann ist offensichtlich auch die Ausgangsgröße beschränkt.

Ein schematisiertes Beispiel für die Faltung ist in Bild 2.3.2 gezeigt. Wie oben diskutiert wurde, ist die Ausgangsgröße „breiter“ als die Eingangsgröße. Die erste Faltung bewirkt eine Verschleifung der Änderungen von $[f_\mu]$, die zweite eine Unterdrückung der konstanten Bereiche beziehungsweise eine Hervorhebung der Änderungen. Ähnliche Ergebnisse werden auch mit anderen ähnlichen Impulsantworten (s. Bild 2.3.8) erreicht.

Die Berechnung der Faltung gemäß (2.3.15) erfordert für $M_x = M_y = M$ und $m_x = m_y = m$ immerhin $\mathcal{O}(M^2 m^2)$ Operationen. Eine deutliche Reduktion der Komplexität ergibt sich, wenn die Impulsantwort $g_{\mu\nu}$ die Bedingung der **Separierbarkeit** erfüllt, nämlich die spezielle

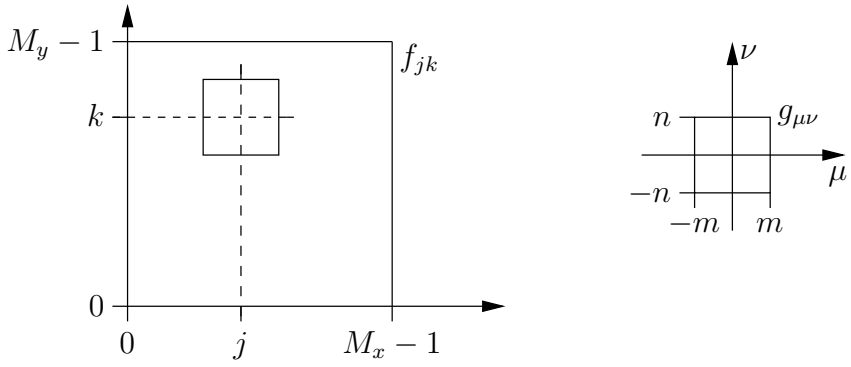


Bild 2.3.3: Faltung als Multiplikation korrespondierender Werte des Musters mit der Maske der Filterkoeffizienten

Form $g_{\mu\nu} = g_\mu g_\nu$ hat. In dem Falle ergibt sich aus (2.3.15)

$$\boxed{\begin{aligned} h_{jk} \;&=\; \sum_{\mu=0}^{m_x-1} \sum_{\nu=0}^{m_y-1} f_{j-\mu,k-\nu} g_\mu g_\nu = \sum_{\mu=0}^{m_x-1} g_\mu \sum_{\nu=0}^{m_y-1} f_{j-\mu,k-\nu} g_\nu \\ &\quad j = 0, 1, \ldots, M_x + m_x - 2\,, \quad k = 0, 1, \ldots, M_y + m_y - 2\,. \end{aligned}} \tag{2.3.18}$$

Die Komplexität dieser Operation ist nur $\mathcal{O}(M^2 m)$. Man wird also immer bestrebt sein, separierbare Impulsantworten zu verwenden, bzw. eine nichtseparierbare durch separierbare zu *approximieren*. Die in (2.3.41) eingeführte GAUSS-Funktion mit ihrer diskreten Approximation g_3 in (2.3.44) ist ein Beispiel für eine separierbare Funktion.

Es ist vielfach zweckmäßig, die Gleichungen für die Faltung so zu modifizieren, dass sie für Funktionen $[g_{jk}]$ (oder $[f_{jk}]$) gelten, die in einem Bereich von Null verschiedene Werte annehmen, der *symmetrisch* um $\mu = \nu = 0$ liegt. Im eindimensionalen Fall hat dann g_j von Null verschiedene Werte für $\boldsymbol{g} = (g_{-m}, \ldots, g_{-1}, g_0, g_1, \ldots, g_m)$. Insbesondere wenn $\boldsymbol{g}$ eine symmetrische Folge von Werten ist, lässt sich die Faltung auch als Positionierung des Zentrums der **Maske** mit den Filterkoeffizienten von $\boldsymbol{g}$ an einer Stelle (j, k) des Musters $\boldsymbol{f}$, Multiplikation korrespondierender Werte im Muster und in der Maske sowie Addition der Produkte auffassen. Dieses zeigt Bild 2.3.3. Die zugehörige Operation ist

$$h_{jk} = \sum_{\mu=-m}^{m} \sum_{\nu=-n}^{n} f_{j-\mu,k-\nu}\, g_{\mu\nu}\;. \tag{2.3.19}$$

2.3.3 Diskrete FOURIER-Transformation

Sowohl für diskrete als auch kontinuierliche Funktionen ist die FOURIER-Transformation ein wichtiges Hilfsmittel bei der Behandlung verschiebungsinvarianter linearer Systeme. Ausgangspunkt ist eine kontinuierliche Funktion $\boldsymbol{f}(\boldsymbol{x})$, die mit Satz 2.1 in eine Folge von Abtastwerten $\boldsymbol{f}$ transformiert wird. Für die weitere Diskussion wird stets von einer zweidimensionalen Funktion $f(x, y)$ und deren Abtastwerten $[f_{jk}]$ ausgegangen, in Bild 2.3.5a ist der Einfachheit halber nur eine eindimensionale Funktion $f(x)$ dargestellt.

Definition 2.6 *Für eine kontinuierliche Funktion $f(x,y)$ ist die* FOURIER-**Transformation** *$F(\xi,\eta)$ definiert durch*

$$F(\xi,\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp[-\mathrm{i}\,(\xi \mathrm{x} + \eta \mathrm{y})]\,\mathrm{d}x\,\mathrm{d}y = \mathrm{FT}\{f(x,y)\}\ . \tag{2.3.20}$$

Die FOURIER-Transformierte existiert, wenn

$$\int_{-\infty}^{\infty} |f(x,y)|\,\mathrm{d}x < \infty \quad \text{und} \quad \int_{-\infty}^{\infty} |f(x,y)|\,\mathrm{d}y < \infty\ . \tag{2.3.21}$$

Zwischen den Funktionen $f(x,y)$ und $F(\xi,\eta)$ besteht ein eindeutig umkehrbarer Zusammenhang, wenn $f(x,y)$ stetig ist. An Unstetigkeitsstellen von $f(x,y)$ sollte man diese durch den Mittelwert ersetzen.

Satz 2.9 *Ist $F(\xi,\eta)$ die durch (2.3.20) definierte* FOURIER*-Transformation einer Funktion $f(x,y)$, so erhält man die Ausgangsfunktion aus dem* **Umkehrintegral**

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\xi,\eta)\exp[\mathrm{i}\,(\xi x + \eta y)]\,\mathrm{d}\xi\,\mathrm{d}\eta = \mathrm{FT}^{-1}\{F(\xi,\eta)\}\ . \tag{2.3.22}$$

Beweis*: s. z. B. [Zygmund, 1968], S. 242–258, [Courant und Hilbert, 1953], II §6, bzw. unter Verwendung der δ–Funktion [Niemann, 1973].*

Wird die Funktion $f(x,y)$ wie in (2.1.1), S. 62, an diskreten Stellen Δx, Δy abgetastet, so entsteht die Folge $[f_{jk}]$ von Abtastwerten. Die Folge der Abtastwerte $[f_{jk}]$ ist vollständig bekannt, wenn man die in (2.1.3) angegebenen $M_x M_y$ Werte f_{jk} kennt. Man kann nun gedanklich die Folge der Abtastwerte in x– und y–Richtung periodisch fortsetzen, indem man

$$f(j+\mu M_x, k+\nu M_y) = f(j,k) = f_{jk} \tag{2.3.23}$$

definiert. Damit erhält man eine Folge von Abtastwerten, die sich periodisch nach M_x Werten in x–Richtung und nach M_y Werten in y–Richtung wiederholt. Somit kann eine endliche Folge von Abtastwerten $[f_{jk}]$ als eine Periode einer gemäß (2.3.23) periodischen Folge aufgefasst werden. Diese periodische Folge wird mit $[\widetilde{f}_{jk}]$ bezeichnet. Der Übergang zwischen beiden Folgen ergibt sich aus

$$\begin{aligned} \widetilde{f}_{j+\mu M_x,k+\nu M_y} &= f_{jk}\ , \quad \mu,\nu = 0,\pm1,\pm2,\ldots \\ f_{jk} &= \widetilde{f}_{jk}\ , \quad \text{jeweils } 0 \le j \le M_x - 1\ , \quad 0 \le k \le M_y - 1\ . \end{aligned} \tag{2.3.24}$$

Der periodischen Folge $[\widetilde{f}_{jk}]$ lässt sich eine periodische Folge $[\widetilde{F}_{\mu\nu}]$ von FOURIER-Koeffizienten zuordnen. Dabei sei $\widetilde{f}_{jk},\ j,k = 0,1,\ldots,M-1$ eine periodische Folge von Abtastwerten im Abstand $\Delta x, \Delta y$ mit Periode x_p, y_p und $\widetilde{F}_{\mu\nu}$ eine periodische Folge von Werten im Abstand $\Delta\xi, \Delta\eta$ mit der Periode ξ_p, η_p, die als FOURIER-Spektrum bezeichnet wird.

Satz 2.10 *Zwischen* FOURIER-*Spektrum und Ausgangsfolge besteht die Beziehung*

$$\widetilde{F}_{\mu,\nu} = \frac{1}{x_p y_p} \sum_{j=0}^{M_x-1} \sum_{k=0}^{M_y-1} \widetilde{f}_{j,k} \exp[-\mathrm{i}\ (\mu\Delta\xi\, j\Delta x + \nu\Delta\eta\, k\Delta y)]\ , \tag{2.3.25}$$

$$\widetilde{f}_{j,k} = \Delta x\Delta y \sum_{\mu=0}^{M_x-1} \sum_{\nu=0}^{M_y-1} \widetilde{F}_{\mu,\nu} \exp[\mathrm{i}\ (\mu\Delta\xi\, j\Delta x + \nu\Delta\eta\, k\Delta y)]\ , \tag{2.3.26}$$

$$\xi_p\Delta x = \eta_p\Delta y = 2\pi\ , \quad \Delta\xi x_p = \Delta\eta y_p = 2\pi\ . \tag{2.3.27}$$

Beweis*: s. z. B. [Oppenheim und Schafer, 1975], S. 87–121.*

Setzt man, wie häufig üblich, $\Delta x = \Delta y = 1$ [Längeneinheit], beachtet $x_p = M_x\Delta x$ und $\xi_p = M_x\Delta\xi$ (s. Bild 2.3.5), so ergibt sich eine spezialisierte Form, die als **diskrete** FOURIER-**Transformation** (DFT) bezeichnet wird. In (2.3.28) oder (2.3.29) wurde davon Gebrauch gemacht, dass bei dem Transformationspaar $f = \mathrm{DFT}\{\mathrm{DFT}^{-1}\{f\}\}$ die Konstanten, nämlich $1/(M_xM_y)$, beliebig auf die beiden Gleichungen verteilt werden können. Es gilt

Satz 2.11 *Definiert man die DFT mit*

$$\widetilde{F}_{\mu\nu} = \sum_{j=0}^{M_x-1} \sum_{k=0}^{M_y-1} \widetilde{f}_{jk} \exp\left[-\mathrm{i}\,2\pi\left(\frac{\mu j}{M_x} + \frac{\nu k}{M_y}\right)\right]$$
$$= \mathrm{DFT}\{[\widetilde{f}_{jk}]\} \qquad \mu,\nu = 0,\pm1,\pm2,\dots\ , \tag{2.3.28}$$

so erhält man die $\widetilde{f}_{jk}$ *aus der inversen Beziehung*

$$\widetilde{f}_{jk} = \frac{1}{M_xM_y} \sum_{\mu=0}^{M_x-1} \sum_{\nu=0}^{M_y-1} \widetilde{F}_{\mu\nu} \exp\left[\mathrm{i}\,2\pi\left(\frac{\mu j}{M_x} + \frac{\nu k}{M_y}\right)\right]$$
$$= \mathrm{DFT}^{-1}\{[\widetilde{F}_{\mu\nu}]\} \qquad j,k = 0,\pm1,\pm2,\dots\ . \tag{2.3.29}$$

Beweis: Die Beweisidee besteht darin zu zeigen, dass $f = \mathrm{DFT}^{-1}\{\mathrm{DFT}\{f\}\}$ ist, indem man (2.3.28) in (2.3.29) einsetzt.

Zur Berechnung der periodischen Folge $[\widetilde{F}_{\mu\nu}]$ reicht eine Periode von $[\widetilde{f}_{jk}]$ aus; das ist einleuchtend, da weitere Perioden keine zusätzliche Information enthalten. Dass $[\widetilde{F}_{\mu\nu}]$ periodisch ist, folgt aus (2.3.28) und der Periodizität der darin auftretenden Exponentialfunktion. Eine analoge Bemerkung trifft für die mit (2.3.29) berechnete Folge $[\widetilde{f}_{jk}]$ zu. Zur Vereinfachung der Notation wird nur für den eindimensionalen Fall gerechnet, also für die Gleichungen

$$\widetilde{F}_{\mu} = \sum_{j=0}^{M-1} \widetilde{f}_j \exp\left[\frac{-\mathrm{i}\,2\pi\mu j}{M}\right]\ ,$$

$$\widetilde{f}_{k} = \frac{1}{M} \sum_{\mu=0}^{M-1} \widetilde{F}_{\mu} \exp\left[\frac{\mathrm{i}\,2\pi\mu k}{M}\right]\ .$$

$$\widetilde{f}_{k} = \frac{1}{M} \sum_{\mu=0}^{M-1} \left(\sum_{j=0}^{M-1} \widetilde{f}_j \exp\left[\frac{-\mathrm{i}\,2\pi\mu j}{M}\right]\right) \exp\left[\frac{\mathrm{i}\,2\pi\mu k}{M}\right]$$

$$= \sum_{j=0}^{M-1} \widetilde{f}_j \sum_{\mu=0}^{M-1} \frac{1}{M} \exp\left[\frac{\mathrm{i}\, 2\pi\mu(k-j)}{M}\right] . \tag{2.3.30}$$

Zunächst wird die Summe

$$S = \frac{1}{M} \sum_{\mu=0}^{M-1} \exp\left[\frac{\mathrm{i}\, 2\pi\mu(k-j)}{M}\right]$$

für sich betrachtet. Man sieht sofort, dass

$$S = 1 \qquad \text{für} \qquad j = k \pmod M \tag{2.3.31}$$

gilt. Für $j \neq k$ lässt sich die Summe in der Form

$$\begin{aligned} S &= \frac{1}{M} \sum_{l=1}^{M} s_l \,, \\ s_l &= \left(\exp\left[\frac{\mathrm{i}\, 2\pi(k-j)}{M}\right]\right)^{l-1} = q^{l-1} \end{aligned}$$

schreiben. Die Summanden bilden also eine geometrische Progression, deren Summe über M Terme bekanntlich

$$\begin{aligned} S &= \frac{1-q^M}{1-q} \\ &= \frac{1-\exp[\mathrm{i}\, 2\pi(k-j)]}{1-\exp\left[\frac{\mathrm{i}\, 2\pi(k-j)}{M}\right]} \end{aligned}$$

ist. Für diese Summe gilt

$$S = 0 \qquad \text{für} \qquad k \neq j \,, \tag{2.3.32}$$

da dann $k - j = \nu \neq 0,\ \exp[\mathrm{i}\, 2\pi\nu] = 1$ und $\exp[\mathrm{i}\, 2\pi\nu/M] \neq 1$ ist. Damit wird in (2.3.30) die Summe über j auf einen von Null verschiedenen Summanden für $j = k$ reduziert, und man erhält

$$\widetilde{f}_k = \widetilde{f}_k \,.$$

Damit ist gezeigt, dass $\widetilde{f}_k = \mathrm{DFT}^{-1}\{\mathrm{DFT}\{\widetilde{f}_k\}\}$ ist. Es sei noch angemerkt, dass man die Konstante $1/(M_x M_y)$ in (2.3.29) einbeziehen kann, wie es hier getan wird. Diese Form wird hier im Hinblick auf Satz 2.12 bevorzugt. Natürlich kann man die Konstante auch in (2.3.28) einbeziehen, oder sie symmetrisch auf beide verteilen. Wenn die Konstante in (2.3.28) einbezogen wird, ergibt sich in Satz 2.12 die Beziehung $H_\nu = M F_\nu G_\nu$. Der Beweis von Satz 2.11 ist damit abgeschlossen.

Satz 2.11 enthält die wichtige Aussage, dass sich einer periodischen Folge von Abtastwerten $[\widetilde{f}_{jk}]$ eines Musters, dem **Ortsbereich** (bzw. dem **Zeitbereich**), eine periodische Folge von FOURIER-**Koeffizienten** $[\widetilde{F}_{\mu\nu}]$, der **Frequenzbereich**, eindeutig umkehrbar zuordnen lässt. Man kann auch sagen, dass eine periodische diskrete Funktion ein periodisches diskretes **Spektrum** hat. Die FOURIER-Koeffizienten der Impulsantwort eines linearen Filters $\boldsymbol{g}$ werden auch

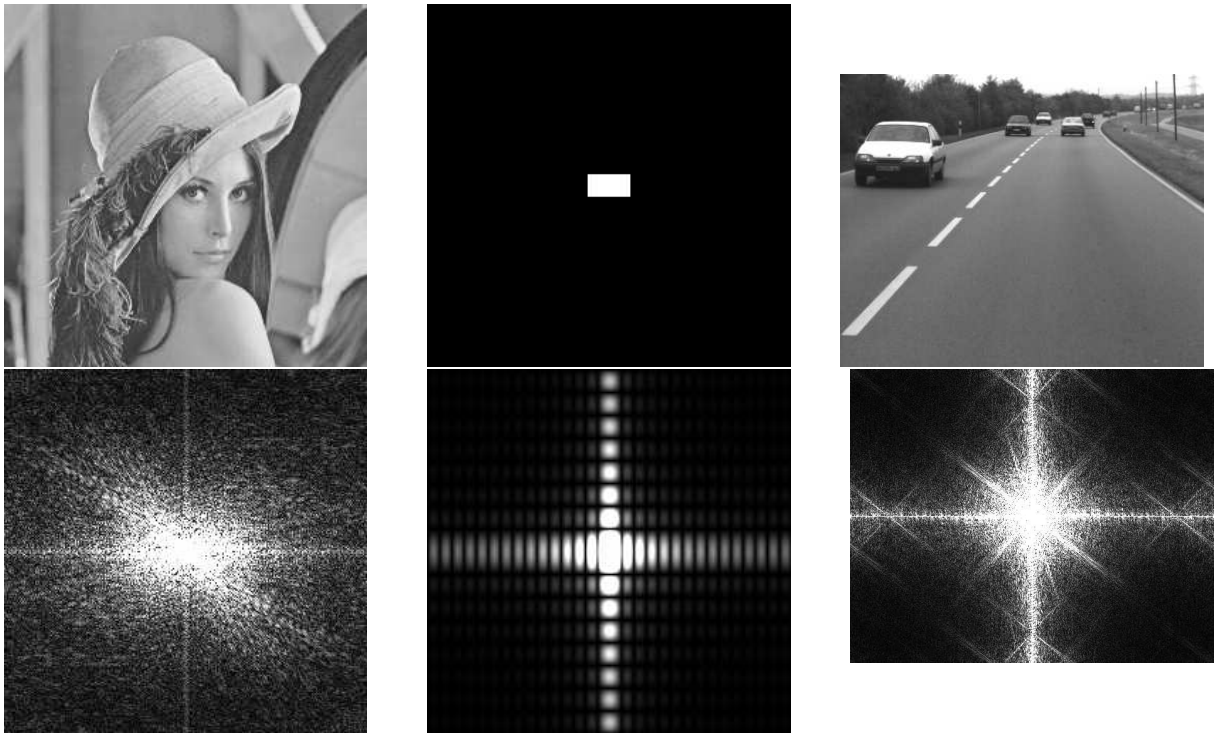


Bild 2.3.4: Jeweils oben ein Bild und unten die zugehörige DFT

als **Frequenzgang** bezeichnet. Bild 2.3.4 zeigt Beispiele für Funktionen und deren diskrete FOURIER-Transformierte.

Bei allen Anwendungen der DFT auf endliche Folgen $[f_{jk}]$ gemäß (2.1.1) ist es äußerst wichtig, stets daran zu denken, dass zumindest gedanklich $[f_{jk}]$ periodisch wiederholt wird. Tatsächlich hat man es also nicht mit endlichen Folgen $[f_{jk}]$ und $[F_{\mu\nu}]$ zu tun, sondern mit unendlich periodischen Folgen $[\widetilde{f}_{jk}]$ und $[\widetilde{F}_{\mu\nu}]$, obwohl man natürlich praktisch immer nur eine Periode betrachten wird. Daraus folgt auch sofort, dass man mit der DFT nur Abtastwerte von Funktionen $f(x, y)$ verarbeiten darf, die gemäß (2.1.18) beziehungsweise (2.1.26) bandbegrenzt sind, da nur dann das Spektrum auf eine Periode begrenzt werden kann. Dieses wird in Bild 2.3.5 verdeutlicht. Es wird noch darauf hingewiesen, dass man (2.3.28) auch in der Form

$$\widetilde{F}_{\mu\nu} = \sum_{j=0}^{M_x-1} \left(\sum_{k=0}^{M_y-1} \widetilde{f}_{jk} \exp\left[\frac{-\mathrm{i}\, 2\pi\nu k}{M_y}\right] \right) \exp\left[\frac{-\mathrm{i}\, 2\pi\mu j}{M_x}\right]$$

schreiben kann. Das bedeutet, dass man eine mehrdimensionale DFT stets auf mehrere eindimensionale DFT zurückführen kann.

Die Bedeutung der DFT liegt in zwei Punkten. Zum einen ist es wegen Satz 3.3 in Abschnitt 3.2.1 mit der *schnellen* FOURIER-*Transformation* möglich, die DFT sehr effizient zu berechnen. Zum anderen bietet die DFT eine weitere Möglichkeit, die Ausgangsgröße eines verschiebungsinvarianten linearen Systems zu berechnen. Die Grundlage dafür bildet der folgende *Multiplikationssatz* der FOURIER-Transformation. Er besagt in Worten, dass einer Multiplikation zweier periodischer diskreter Folgen im Frequenzbereich deren zyklische Faltung im Ortsbereich entspricht. Dieser Satz gilt analog auch für kontinuierliche Funktionen.

Es seien $[\widetilde{f}_{jk}]$ und $[\widetilde{g}_{jk}]$ zwei periodische Folgen mit der gemeinsamen Periodenlänge M'_x, M'_y, die noch geeignet festzulegen ist. Für diese werden mit der DFT die periodischen

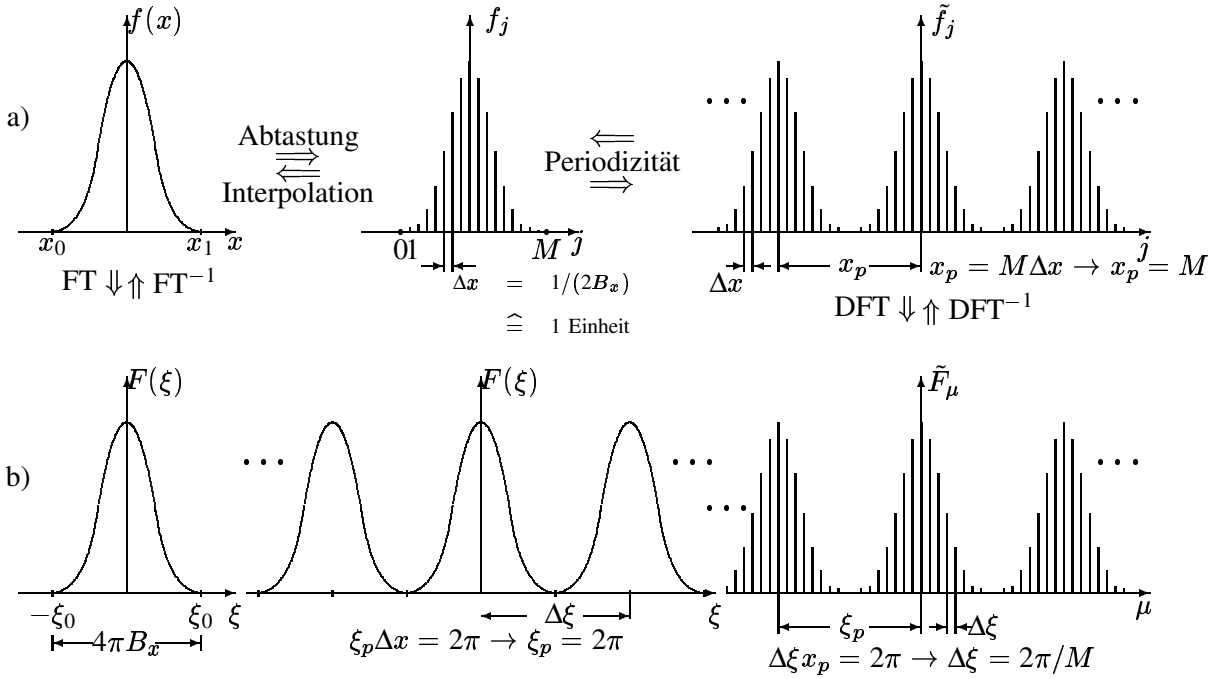


Bild 2.3.5: Eine kontinuierliche Funktion $f(x)$ hat ein kontinuierliches Spektrum $F(\xi)$, eine periodische diskrete Funktion $[\widetilde{f}_j]$ hat ein periodisches diskretes Spektrum $[\widetilde{F}_\mu]$. Die periodische Fortsetzung ist mit drei Punkten angedeutet, der Ortsbereich in a), der Frequenzbereich in b) dargestellt.

Folgen $[\widetilde{F}_{\mu\nu}]$ und $[\widetilde{G}_{\mu\nu}]$ gemäß (2.3.28) berechnet. Damit wird eine periodische Folge $[\widetilde{H}_{\mu\nu}]$ definiert durch

$$\begin{aligned} [\widetilde{H}_{\mu\nu}] &= [\widetilde{F}_{\mu\nu}][\widetilde{G}_{\mu\nu}] \,, \\ \widetilde{H}_{\mu\nu} &= \widetilde{F}_{\mu\nu}\widetilde{G}_{\mu\nu} \,, \qquad \mu,\nu = 0,\pm1,\pm2,\ldots \,. \end{aligned} \tag{2.3.33}$$

Für $\widetilde{H}$ gilt die folgende Aussage:

Satz 2.12 *(***Multiplikationssatz** *der DFT) Die periodische Folge*

$$\widetilde{h}_{jk} = \mathrm{DFT}^{-1}\{[\widetilde{H}_{\mu\nu}]\} \tag{2.3.34}$$

ist gleich dem Ergebnis der **zyklischen Faltung** *von* $\widetilde{f}_{jk}$ *und* $\widetilde{g}_{jk}$

$$\widetilde{h}_{jk} = \sum_{\mu=0}^{M'_x-1} \sum_{\nu=0}^{M'_y-1} \widetilde{f}_{\mu\nu}\widetilde{g}_{j-\mu,k-\nu} \,, \qquad j,k = 0,\pm1,\ldots \tag{2.3.35}$$

In einer Gleichung zusammengefasst erhält man also die gemäß (2.3.34) definierte periodische Folge aus

$$\widetilde{h}_{jk} = \mathrm{DFT}^{-1}\{\mathrm{DFT}\{[\widetilde{f}_{jk}]\} \cdot \mathrm{DFT}\{[\widetilde{g}_{jk}]\}\} \,. \tag{2.3.36}$$

Beweis: Die Beweisidee besteht einfach darin, (2.3.36) direkt auszuwerten, d. h. man setzt (2.3.33) in (2.3.34) ein. Der Beweis wird hier der Einfachheit halber nur für Folgen $[\widetilde{f}_j], [\widetilde{g}_j], [\widetilde{h}_j]$ geführt. Mit (2.3.28) ist

$$\begin{aligned}
\widetilde{F}_\mu &= \sum_{j=0}^{M'_x-1} \widetilde{f}_j \exp\left[\frac{-\mathrm{i}\,2\pi\mu j}{M'_x}\right] , \\
\widetilde{G}_\mu &= \sum_{k=0}^{M'_x-1} \widetilde{g}_k \exp\left[\frac{-\mathrm{i}\,2\pi\mu k}{M'_x}\right] , \qquad \mu = 0, \pm 1, \pm 2, \ldots \\
\widetilde{F}_\mu \widetilde{G}_\mu &= \sum_j \sum_k \widetilde{f}_j \widetilde{g}_k \exp\left[\frac{-\mathrm{i}\,2\pi\mu(j+k)}{M'_x}\right] .
\end{aligned}$$

Aus (2.3.34) und (2.3.29) folgt

$$\begin{aligned}
\widetilde{h}_l &= \frac{1}{M'_x} \sum_{\mu=0}^{M'_x-1} \sum_j \sum_k \widetilde{f}_j \widetilde{g}_k \exp\left[\frac{\mathrm{i}\,2\pi\mu(l-j-k)}{M'_x}\right] \\
&= \sum_j \widetilde{f}_j \sum_k \widetilde{g}_k \frac{1}{M'_x} \sum_\mu \exp\left[\frac{\mathrm{i}\,2\pi\mu(l-j-k)}{M'_x}\right] .
\end{aligned}$$

Ein Vergleich mit (2.3.31), (2.3.32) ergibt, dass der Term

$$\frac{1}{M'_x} \sum_\mu \exp\left[\frac{\mathrm{i}\,2\pi\mu(l-j-k)}{M'_x}\right] = 1 \qquad \text{für} \qquad k = l - j$$

ergibt und sonst Null ist, sodass

$$\widetilde{h}_l = \sum_{j=0}^{M'_x} \widetilde{f}_j \, \widetilde{g}_{l-j} , \qquad l = 0, \pm 1, \pm 2, \ldots \quad .$$

Damit ist Satz 2.12 bewiesen.

Aus Bild 2.3.6 und (2.3.14) wird klar, dass Satz 2.12 dann zur Berechnung der Ausgangsgröße eines verschiebungsinvarianten linearen Systems herangezogen werden kann, wenn

$$M'_x \geq M_x + m_x - 1 , \qquad M'_y \geq M_y + m_y - 1 \tag{2.3.37}$$

ist, da dann das Ergebnis von einer Periode der zyklischen Faltung gleich dem Ergebnis der diskreten Faltung ist; man erhält die Folge $[h_{jk}]$ aus einer Periode von $[\widetilde{h}_{jk}]$. Wenn $M_x M_y$ und $m_x m_y$ wie in (2.3.14) die Zahl der von Null verschiedenen Werte f_{jk} und g_{jk} ist, so lässt sich (2.3.37) stets dadurch sicherstellen, dass man f_{jk} und g_{jk} durch Nullen auffüllt ("zero padding"). Offensichtlich ist die Anwendung von (2.3.36) auf Funktionen $\boldsymbol{f}, \boldsymbol{g}$ von endlicher Ausdehnung beschränkt, da nur dann (2.3.37) eingehalten werden kann ("finite impulse response" FIR). Zu Systemen mit Impulsantworten von unendlicher Ausdehnung ("infinite impulse response" IIR) wird auf die Literatur verwiesen, da deren Behandlung hier zu weit führen würde.

Die Berechnung der Faltung im Frequenzbereich, also mit (2.3.36), statt der Berechnung im Ortsbereich, also mit (2.3.14), (2.3.15) scheint zunächst numerisch nicht attraktiv zu sein, da man im Frequenzbereich drei Doppelsummen auswerten muss, im Ortsbereich nur eine. Die numerische Berechnung der DFT erfolgt jedoch nicht durch „naive" Implementierung der Doppelsumme nach (2.3.28), sondern über die *schnelle* FOURIER-*Transformation* ("Fast FOURIER

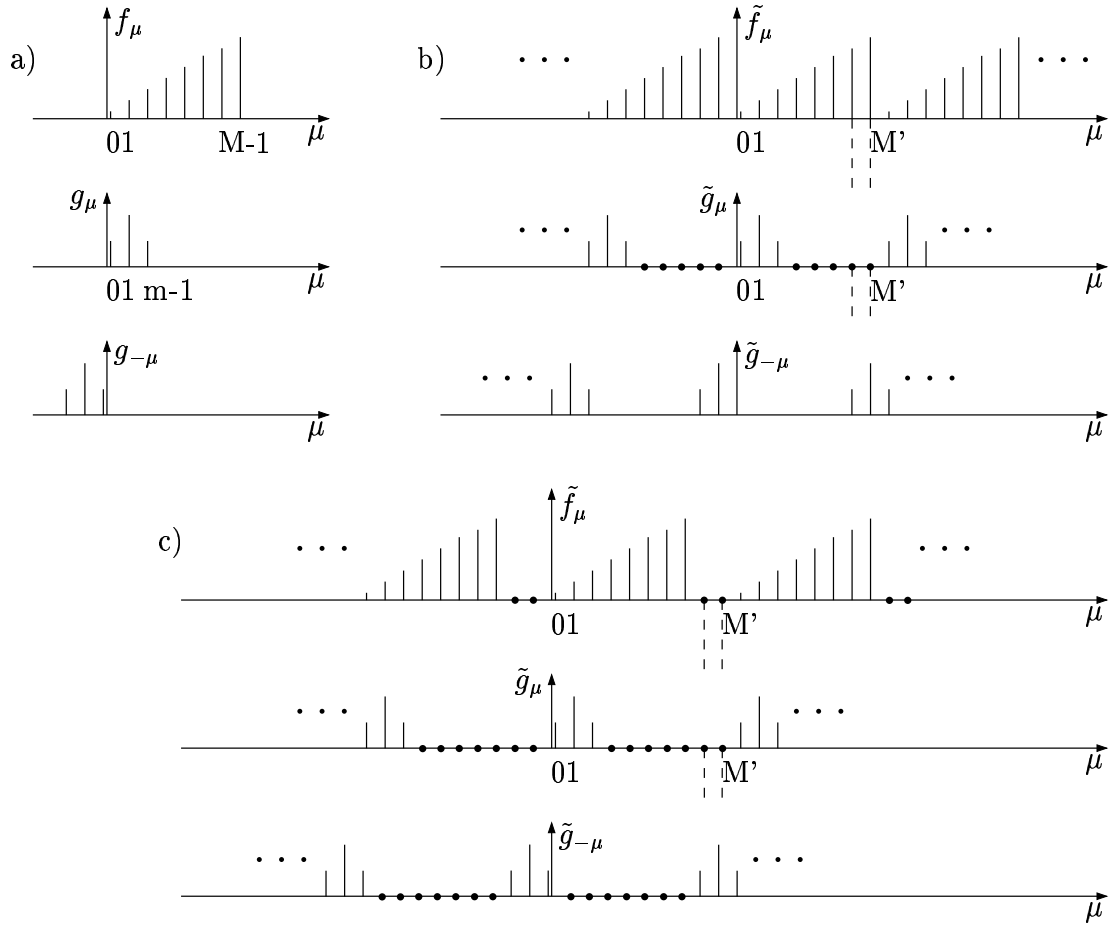


Bild 2.3.6: In a) ist die diskrete Faltung gemäß (2.3.14) angedeutet, in b) die zyklische Faltung gemäß (2.3.35), wobei f_μ und g_μ mit (2.3.24) periodisch fortgesetzt wurden. Man erkennt, dass bei diesem Wert von M' eine Periode des Ergebnisses der zyklischen Faltung nicht mit dem Ergebnis der diskreten Faltung übereinstimmt. In c) wurde gemäß (2.3.37) $M' = M + m - 1$ gewählt, und nun ist das Ergebnis der diskreten Faltung identisch mit einer Periode des Ergebnisses der zyklischen Faltung. Die Ergebnisse der Faltungen wurden jedoch nicht dargestellt.

Transform", FFT). Ihre Basis ist die in Satz 3.3, S. 173, angegebene Faktorisierung der Transformationsmatrix, und zu weiteren Einzelheiten wird auf die Literatur verwiesen. Die Faltung im Frequenzbereich ist dann aus Komplexitätsgründen vorteilhaft, wenn die Ausdehnung von $[g_{jk}]$ genügend groß ist. Eine genaue Aussage ist nur bei Kenntnis des Verhältnisses von Additions- und Multiplikationszeiten möglich; als Anhaltspunkt kann gelten, dass (2.3.15) (bzw. (2.3.19)) etwa bis $m_x \times m_y = 7 \times 7$ oder 9×9 vorzuziehen ist. Insbesondere wenn man nur relativ einfache Muster betrachtet, wie es bei Klassifikationsaufgaben oft der Fall ist, wird man vielfach mit Impulsantworten recht kleiner Ausdehnung auskommen.

Die Beeinflussung eines Signals durch ein lineares System wird als lineare Filterung – oder wenn keine Verwechslung möglich ist auch kurz als Filterung – bezeichnet, das lineare System heißt auch **lineares Filter** oder kurz Filter. Die Synthese von Filtern mit vorgegebenen Eigenschaften ist ein aus der Nachrichtentechnik wohlbekanntes Problem. Es ist zu betonen, dass für Zwecke der Mustererkennung die Synthese eines Filters, z. B. mit vorgegebener Dämpfung und Phase, nicht im Vordergrund steht, da bisher keine Ergebnisse vorliegen, welche darauf hindeu-

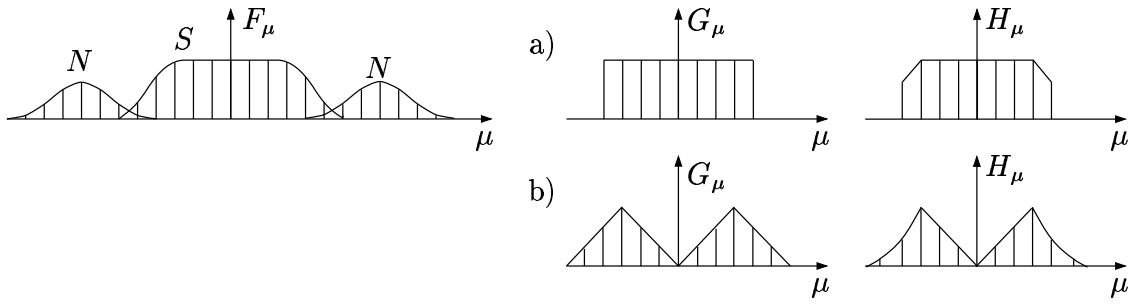


Bild 2.3.7: a) Prinzip der Störungsreduktion durch Filterung. b) Hervorhebung bestimmter Frequenzbereiche

ten, dass dieses für die Leistung eines Klassifikationssystems wichtig ist. In diesem Zusammenhang wird an die Ausführungen zu Beginn von Kapitel 2 erinnert, in denen die Problematik der Beurteilung von Vorverarbeitungsoperationen erörtert wurde.

2.3.4 Gesichtspunkte zur Auswahl eines linearen Systems

Nachdem geklärt ist, wie man die Ausgangsgröße eines verschiebungsinvarianten linearen Systems berechnet, bleibt nun noch das Problem, die Impulsantwort eines Systems für die Vorverarbeitung von Mustern festzulegen. Dafür gibt es leider nur wenige allgemeine Gesichtspunkte, aber viele spezielle Einzelergebnisse. Für Zwecke der Mustererkennung, insbesondere von Bildern, sind Separierbarkeit, die Vermeidung von negativen Ausgabewerten bei nur nichtnegativen Eingabewerten sowie u. U. Isotropie (d. h. Richtungsunabhängigkeit der Ausgabe) und Vermeidung von Verschiebungen der Ausgabe relativ zur Eingabe nützliche Anforderungen. Ein allgemeines Modell für das aufgenommene Muster $\boldsymbol{f}$ ist

$$\boldsymbol{f} = \boldsymbol{s} * \boldsymbol{g} + \boldsymbol{n} \; . \tag{2.3.38}$$

Es geht davon aus, dass ein „ideales“ Muster $\boldsymbol{s}$ durch die Einwirkung eines linearen Systems verzerrt und das Ergebnis dieser Faltung noch durch einen additiven Störprozess $\boldsymbol{n}$ beeinträchtigt wird. Gesucht ist i. Allg. ein lineares System $\boldsymbol{\gamma}$, welches gemäß

$$\boldsymbol{h} = \boldsymbol{f} * \boldsymbol{\gamma} = \widehat{\boldsymbol{s}} \simeq \boldsymbol{s} \tag{2.3.39}$$

als Ausgangsgröße $\boldsymbol{h}$ eine möglichst gute Approximation $\widehat{\boldsymbol{s}}$ an das ideale Muster $\boldsymbol{s}$ ergibt. Dieser allgemeine Fall der *Restauration*, wird hier nicht behandelt sondern der Leser auf die Literatur verwiesen. Hier werden nur zwei einfache Spezialfälle erwähnt:

1. Störungen im Muster sind zu reduzieren.
2. Wichtige Anteile im Muster sind hervorzuheben.

Meistens wird subjektiv beurteilt, ob $\boldsymbol{h}$, und damit $\boldsymbol{\gamma}$, „genügend gut“ ist. In beiden Fällen geht man davon aus, dass der Einfluss von $\boldsymbol{g}$ in (2.3.38) vernachlässigbar ist, dass also $g_{jk} \simeq \delta_{jk}$ ist.

Die Reduzierung einer additiven Störung ist im Prinzip einfach, vorausgesetzt dass die in Bild 2.3.7a gezeigten Verhältnisse zumindest näherungsweise zutreffen. Die Spektren von Signal $\boldsymbol{s}$ und Störung $\boldsymbol{n}$ sind hier einigermaßen getrennt. In der Tat sind Störungen oft relativ hochfrequent im Vergleich zu den Signalanteilen. Man sieht sofort, dass ein Filter $\boldsymbol{\gamma}$, welches

hauptsächlich die niedrigen Frequenzanteile des Signals passieren lässt, eine Reduzierung der Störung bewirkt. Je mehr sich die Spektren von Signal und Störung überlappen, umso geringer ist der Erfolg bei der Störungsreduktion, da man dann mit einem Filter nicht nur Frequenzanteile der Störung sondern auch des Signals beeinflusst. Ein Filter, das vor allem niederfrequente Anteile eines Eingangssignals passieren lässt, wird als **Tiefpass** bezeichnet. Zur Störungsreduktion bei Sprache wird zudem auf Abschnitt 2.5.7 verwiesen.

Als wichtige Anteile in einem Muster gelten oft solche, deren Frequenzanteile im Spektrum relativ hoch liegen. Typische Beispiele sind Ecken und Kanten in einem Bild sowie bei Sprache die Formanten, die im Vergleich zur Sprachgrundfrequenz ebenfalls relativ hochfrequent sind. Ein geeignetes Filter ist hier also ein **Hochpass**, der vor allem die hochfrequenten Anteile passieren lässt. Er würde allerdings auch etwa vorhandene (im Vergleich zum Signal) hochfrequente Störungen passieren lassen. Daher ist i. Allg. ein **Bandpass**, der nur ein bestimmtes Frequenzband passieren lässt, geeigneter. Es wird also in solchen Fällen ein System mit einem Frequenzgang G_μ gemäß Bild 2.3.7b verwendet. Das durchgelassene Frequenzband muss so gewählt werden, das die gewünschten wichtigen Anteile in einem Muster passieren können.

Natürlich kann der genauere Verlauf des Frequenzganges, also der FOURIER-Transformierten der Impulsantwort des linearen Systems, nur bei Kenntnis der ungefähren Spektren von Signal und Störung festgelegt werden. Die „einfache aber ziemlich naive“ Vorgehensweise besteht darin, die Koeffizienten G_μ anhand des erwünschten Frequenzgangs zu bestimmen, wie es auch in Bild 2.3.7 angedeutet ist, und als Impulsantwort h_j eine Periode der inversen DFT der Koeffizienten G_μ zu verwenden, wobei zur Vereinfachung der Faltung (2.3.14) vielfach auch nur die ersten wenigen Werte von h_j verwendet und alle anderen, insbesonere relativ kleine, Null gesetzt werden sowie u. U. auch noch reelle Werte auf ganzzahlige gerundet werden. Wenn die Übertragungsfunktion eine Unstetigkeit enthält (wie G_μ in Bild 2.3.7a), so sind die Filterkoeffizienten noch mit einer geeigneten *Fensterfunktion* zu multiplizieren; Beispiele dafür gibt (2.5.43), S. 131. Diese einfache Vorgehensweise erlaubt eine erste Näherung an einen gewünschten Frequenzgang und muss hier aus Platzgründen genügen. Aufwendigere und leistungsfähigere Verfahren der Filtersynthese sind in der zitierten Literatur zu finden. Bild 2.3.2 gibt ein schematisiertes Beispiel für die Wirkung einfacher Filter. Mit dem ersten (Tiefpass) werden Änderungen im Muster $\boldsymbol{f}$ reduziert, mit dem zweiten (Hochpass) werden sie hervorgehoben.

2.3.5 Beispiele für lineare Filter

In Bild 2.3.8 sind einige Beispiele für Folgen und deren diskrete FOURIER-Transformierte gezeigt; die Funktion in Bild 2.3.8b oder k rechts entspricht etwa g_μ in Bild 2.3.2 oben, und die in Bild 2.3.8c rechts in etwa g_μ in Bild 2.3.2 unten. Für die Anwendung sei nochmals an (2.3.37) erinnert. Eine Operation, die im Wesentlichen einen **Mittelwert** zwischen dem Bildpunkt und einer kleinen Nachbarschaft berechnet (eindimensionaler Fall s. Bild 2.3.8b,k), betont die tiefen Frequenzen (Tiefpass) und führt zu einer Reduzierung kleiner Störstellen im Bild, zur Verschleifung von Grauwertänderungen und allgemein zu einer **Glättung** des Bildes. Eine Operation, die im Wesentlichen die Differenz zwischen dem Bildpunkt und einer kleinen Nachbarschaft von z. B. 3×3 Bildpunkten bildet (eindimensionaler Fall s. Bild 2.3.8c), betont demnach die hohen Frequenzen (Hochpass) und führt zu einer Hervorhebung von Grauwertänderungen bzw. zu einer **Kontrastverstärkung** im Bild. Ein Bandpass ergibt sich durch Kombination von Tief– und Hochpass.

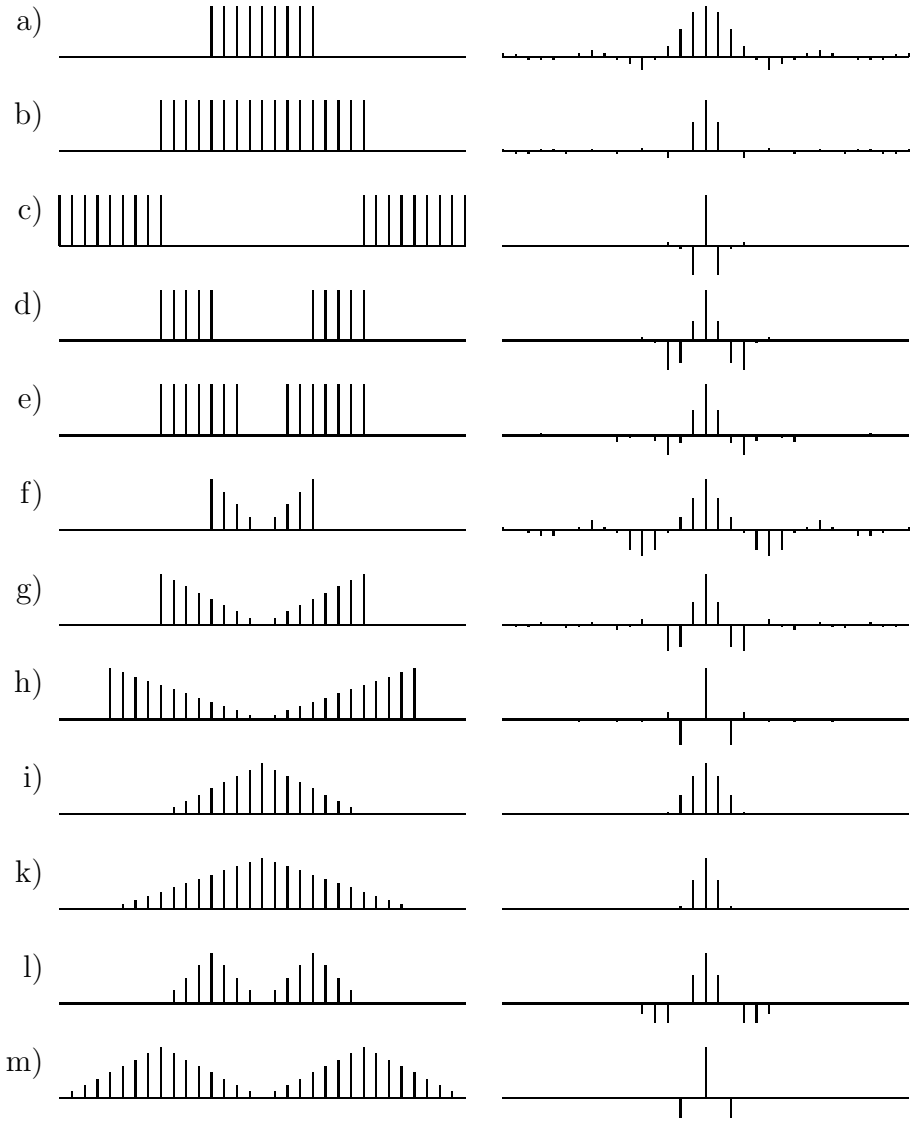


Bild 2.3.8: Einige Beispiele für Folgen, die über die DFT zusammenhängen. Es wurde mit $M = 33$ gerechnet, Funktionswerte identisch oder nahe bei Null wurden nicht dargestellt. Die FOURIER-Koeffizienten sind jeweils links dargestellt

Eine einfache Operation zur **Störungsreduktion** ist demnach die Mittelung

$$h_{jk} = \alpha \sum_{\mu=-m}^{m} \sum_{\nu=-n}^{n} f_{j+\mu,k+\nu} \ , \tag{2.3.40}$$

wobei oft zur Reduktion der Rechenkomplexität $m = n = 1$, also eine 3×3 Nachbarschaft, und $\alpha = 1$ oder $\alpha = 1/((2m+1)(2n+1))$ gewählt wird. Ein anderes Beispiel für einen Tiefpass ist die GAUSS-Funktion bzw. das GAUSS-**Filter**

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] = \mathcal{N}(x,y|0,\sigma) \ , \tag{2.3.41}$$

dessen Durchlassbereich durch den Parameter σ eingestellt werden kann, wie auch Bild 4.2.1, S. 324, für den eindimensionalen Fall zeigt. Ein kleiner Wert von σ ergibt eine *schmale* Im-

pulsantwort, daher wegen Bild 2.1.4, S. 68, einen *breiten* Durchlassbereich des Filters und eine geringe Glättung. Das zeigt auch die FOURIER-Transformierte der GAUSS-Funktion

$$G(\xi,\eta) = \exp\left[-\frac{\sigma^2}{2}\left(\xi^2+\eta^2\right)\right] , \tag{2.3.42}$$

die selbst wieder eine GAUSS-Funktion ist; die eindimensionale Version wurde bereits in (2.1.8), S. 64, eingeführt. Die anisotrope Version, die zur Glättung von Linienstrukturen geeignet ist, wird in (2.3.69) erwähnt. Man erhält die diskrete (eindimensionale) Impulsantwort nach dem oben genannten Verfahren aus

$$g_j = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{j^2}{2\sigma^2}\right] , \quad j = \ldots,-2,-1,0,1,2,\ldots , \tag{2.3.43}$$

wobei wie in Abschnitt 2.1.2 die Breite und Höhe eines Bildpunktes gleich Eins gesetzt wurde. Wegen der Separierbarkeit der GAUSS-Funktion ist damit auch der n–dimensionale Fall abgedeckt. Eine effiziente rekursive Realisierung wird in Abschnitt 2.3.6 angegeben.

Einige Beispiele für Impulsantworten von Glättungsfiltern sind

$$g_1 = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} , \quad g_2 = \frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} , \quad g_3 = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} . \tag{2.3.44}$$

Diese sind symmetrische Impulsantworten, d. h. der Wert $g_{i,00}$ liegt jeweils in der Mitte der Matrix. Die Impulsantworten g_1, g_3 sind offensichtlich separierbar im Sinne von (2.3.18). Die eindimensionale Version von g_3 ist z. B. $g_{3,1} = (1/4)(1,\ 2,\ 1)$, und die zweidimensionale Version erhält man aus $g_3 = g_{3,1}^{\mathsf{T}} g_{3,1}$.

Wenn man das beobachtete Muster ${}^{\varrho}\boldsymbol{f}$ darstellen kann durch

$${}^{\varrho}\boldsymbol{f} = \boldsymbol{s} + {}^{\varrho}\boldsymbol{n} \tag{2.3.45}$$

und wenn es möglich ist, mehrere Realisationen von ${}^{\varrho}\boldsymbol{f}$, $\varrho = 1,\ldots,N$ zu beobachten, welche das gleiche Signal $\boldsymbol{s}$ und verschiedene Repräsentanten ${}^{\varrho}\boldsymbol{n}$ des gleichen Störprozesses mit $E\{{}^{\varrho}\boldsymbol{n}\} = 0$ enthalten, dann ist es möglich, die Störung durch eine Mittelung

$$\boldsymbol{h} = \frac{1}{N}\sum_{\varrho=1}^{N} {}^{\varrho}\boldsymbol{f} \approx \boldsymbol{s} \tag{2.3.46}$$

über die beobachteten Muster zu reduzieren. Dieser Fall tritt vor allem bei periodisch sich wiederholenden Vorgängen auf, wie z. B. im Geräusch einer rotierenden Maschine oder im Strahlungsbild des schlagenden Herzens bei nuklearmedizinischen Aufnahmen. Natürlich muss die Aufnahme von ${}^{\varrho}\boldsymbol{f}$ genau mit dem periodischen Vorgang synchronisiert werden.

Zur Hervorhebung von Änderungen eignet sich eine Differenzbildung (ROBERTS-Kreuz) gemäß

$$\begin{aligned} h_{jk} &= \sqrt{f_x^2 + f_y^2} \qquad \text{oder} \quad h_{jk} = |f_x| + |f_y| , \\ f_x &= f_{jk} - f_{j+1,k+1} \quad \text{und} \quad f_y = f_{j,k+1} - f_{j+1,k} . \end{aligned} \tag{2.3.47}$$

Eine Alternative ist die diskrete Approximation der partiellen Ableitungen an der Stelle (j,k) durch

$$\frac{\partial f}{\partial x} \approx f_x = \frac{1}{2}(f_{j+1,k} - f_{j-1,k}) , \quad \frac{\partial f}{\partial y} \approx \frac{1}{2}(f_{j,k+1} - f_{j,k-1}) \tag{2.3.48}$$

oder durch den SOBEL-**Operator** an der Stelle (j,k)

$$\begin{aligned}\frac{\partial f}{\partial x} &\approx (f_{j+1,k-1}+2f_{j+1,k}+f_{j+1,k+1})-(f_{j-1,k-1}+2f_{j-1,k}+f_{j-1,k+1})\,,\\ \frac{\partial f}{\partial y} &\approx (f_{j-1,k+1}+2f_{j,k+1}+f_{j+1,k+1})-(f_{j-1,k-1}+2f_{j,k-1}+f_{j+1,k-1})\,.\end{aligned} \tag{2.3.49}$$

Dieser Operator zeigt wegen der Einbeziehung von sechs Bildpunkten eine geringere Empfindlichkeit gegenüber Rauschen.

Andere Approximationen der partiellen Differentiation nach x und y als mit erhält man mit

$$g_5 = \frac{1}{8}\,(1,\,-6,\,0,\,6,\,-1)\;,\quad g_6 = \frac{1}{32}\,(-2,\,9,\,-28,\,0,\,28,\,-9,\,2)\;. \tag{2.3.50}$$

Die partielle Differentiation nach y erfolgt mit der gleichen Impulsantwort um 90^o gedreht. Die Approximation in (2.3.48) entspricht offenbar der Faltung mit $g_4 = 1/2(-1,\,0,\,1)$. Die Faltung entspricht hier direkt der Anwendung der Masken g_i auf das Bild, wie es in (2.3.19) und Bild 2.3.3 angegeben ist. Impulsantworten für die Approximation partieller Ableitungen zweiter Ordnung $\partial^2 f/\partial x^2$ bzw. $\partial^2 f/\partial y^2$ sind z. B.

$$g_7 = \frac{1}{12}(-1,\,16,\,-30,\,16,\,-1) \tag{2.3.51}$$

und für die partielle Ableitung $\partial^2 f/\partial x\,\partial y$

$$g_8 = g_{8,s}^{\mathsf{T}} g_{8,s}\;,\quad g_{8,s} = \frac{1}{12}(1,\,-8,\,0,\,8,\,-1)\;. \tag{2.3.52}$$

Eine Hervorhebung von Änderungen im Bild leistet auch die Operation

$$h_{jk} = (f_{j,k-1}+f_{j,k+1}+f_{j+1,k}+f_{j-1,k})-4f_{jk}\;, \tag{2.3.53}$$

die eine diskrete Version des LAPLACE-**Operators**

$$\begin{aligned}h(x,y) &= \frac{\partial^2 f(x,y)}{\partial x^2}+\frac{\partial^2 f(x,y)}{\partial y^2}\\ &= \nabla^2 f(x,y)\end{aligned} \tag{2.3.54}$$

ist. Konstante Bereiche im Muster werden durch (2.3.47) – (2.3.53) völlig unterdrückt. Eine Modifikation ist die Operation

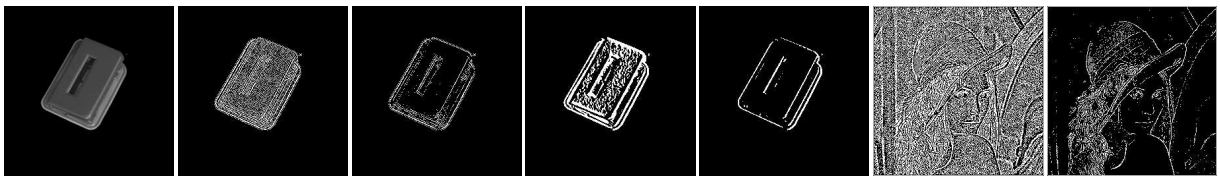
$$h_{jk} = (1+4\alpha)f_{jk}-\alpha(f_{j,k-1}+f_{j,k+1}+f_{j+1,k}+f_{j-1,k})\;, \tag{2.3.55}$$

welche ebenfalls eine Kontrastverbesserung liefert und im kontinuierlichen Fall der Operation

$$h(x,y) = f(x,y)-\alpha\nabla^2 f(x,y) \tag{2.3.56}$$

entspricht.

Einige Beispiele für die Wirkung von Hochpassfiltern zeigt Bild 2.3.9. Für die Tiefpassfilter werden keine Beispiele gezeigt, da sie im Wesentlichen eine Verschleifung von Konturen bzw. ein unschärferes Bild bewirken. Die Wirkung von drei Hochpässen, nämlich den Hochpass in Bild 2.3.2 mit der Impulsantwort $g_x = (-0,5,\,1,\,-0,5)$ in x-Richtung und entsprechend in y-Richtung , den SOBEL-Operator in (2.3.49) und den LAPLACE-Operator in (2.3.54), wird für zwei Originalbilder, nämlich den Locher in Bild 2.3.9a) und Lena in Bild 2.3.4, gezeigt. Das



a) b) c) d) e) f) g)

Bild 2.3.9: Das Bild zeigt: a) Originalbild eines Lochers; b) Locher nach Faltung mit dem Hochpass in Bild 2.3.2; c) Binärbild der 2% größten Werte; d) Locher nach Faltung mit dem SOBEL-Operator; e) Binärbild der 2% größten Werte; f) Lena (Original links oben in Bild 2.3.4) nach Faltung mit dem LAPLACE-Operator; g) Binärbild der 10% größten Werte;

Bild zeigt jeweils zum Einen das Ergebnis direkt, zum Anderen die 2% der größten Werte für den Locher und die 10% der größten Werte für Lena.

Da eine Ableitung, noch mehr eine zweite Ableitung, als Hochpassfilter wirkt (s. (2.1.11)), werden damit hochfrequente Störungen hervorgehoben. Um dem entgegenzuwirken, ist also eine vorherige Tiefpassfilterung zweckmäßig. Wenn diese mit einem GAUSS-Filter erfolgt, können Ableitungen auch effizient rekursiv berechnet werden, wie es in Abschnitt 2.3.6 kurz beschrieben wird.

Einige weitere Beispiele für Impulsantworten zur Hervorhebung von hohen Frequenzen in Bildern, d. h. von Grauwertänderungen, sind

$$g_9 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad g_{10} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix},$$

$$g_{11} = \frac{1}{9}\begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}. \tag{2.3.57}$$

Wie erwähnt ergibt die Kombination von Tief– und Hochpass einen Bandpass. Ein Beispiel dafür ist die Glättung mit der GAUSS-Funktion und die Anhebung hoher Frequenzen mit dem LAPLACE-Operator; dieses wird als **LoG–Filter** ("Laplacian of Gaussians") bezeichnet. Die Impulsantwort ist demnach

$$\begin{aligned} h(x,y) &= \nabla^2 \left(f(x,y) * g(x,y)\right) \\ &= \left(\nabla^2 g(x,y)\right) * f(x,y)\,. \end{aligned} \tag{2.3.58}$$

Die letzte Zeile der obigen Gleichung folgt aus der Linearität der Operationen Faltung und Differentiation. Daher reduziert sich der Rechenaufwand erheblich, da das Muster $f(x,y)$ nur mit *einer* Funktion gefaltet werden muss, nämlich mit

$$\begin{aligned} \nabla^2 g(x,y) &= \frac{x^2+y^2-2\sigma^2}{2\pi\sigma^6}\exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \\ &= \psi(r)\,, \quad r = \sqrt{x^2+y^2}\,. \end{aligned} \tag{2.3.59}$$

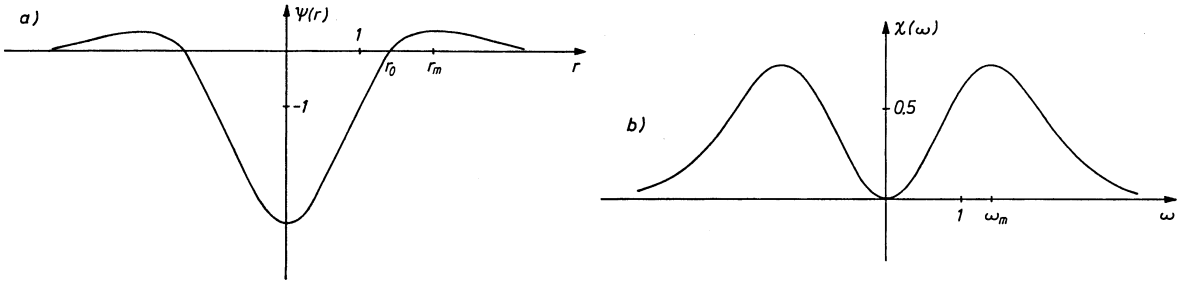


Bild 2.3.10: Ein Beispiel für einen Bandpass durch Kombination von Tiefpass (GAUSS-Funktion) und Hochpass (LAPLACE-Operator).

Die FOURIER-Transformierte von $\nabla^2 g(x,y)$ ist

$$\begin{aligned} \mathrm{FT}\{\nabla^2 g(x,y)\} &= -(\xi^2+\eta^2)\exp\left[-\frac{\sigma^2(\xi^2+\eta^2)}{2}\right] \\ &= \chi(\omega)\,, \quad \omega = \sqrt{\xi^2+\eta^2}\,. \end{aligned} \tag{2.3.60}$$

Die Funktionen $\psi(r)$ und $\chi(\omega)$ enthalten nur den einen Parameter σ, mit dem der Durchlassbereich des Bandpasses eingestellt wird. Sein Maximum liegt bei $|\omega_0| = \sqrt{2}/\sigma$. Diese Funktion ist *nicht* separierbar, lässt sich jedoch durch die Differenz zweier (separierbarer) GAUSS-Funktionen mit unterschiedlicher Streuung σ approximieren; dadurch reduziert sich der Rechenaufwand nochmals. Dieses wird als **DoG–Filter** ("Difference of Gaussians") bezeichnet. Bild 2.3.10 zeigt den Verlauf der Funktionen $\psi(r)$ und $\chi(\omega)$. Wenn man Bild 2.3.10a an der r–Achse umklappt, ähnelt sie einem Mexikanerhut und wird daher z. T. auch so bezeichnet.

2.3.6 Approximation durch rekursive Filter

Gerade bei der Filterung von mehrdimensionalen Mustern spielt die Rechenzeit eine wichtige Rolle. Verbesserungen, die nur auf effizienteren Algorithmen beruhen (und nicht z. B. auf Optimierung des Programmkodes oder Verwendung von Parallelrechnern), sind insbesondere die oben bereits erwähnten Methoden der schnellen FOURIER-Transformation sowie der Separierung und Kaskadierung von Filtern; eine weitere wichtige Methode ist die Approximation eines auf diskreter Faltung gemäß (2.3.15) beruhenden Filters durch ein **rekursives Filter**. Ein solches ist im eindimensionalen Fall gegeben durch die Gleichung

$$\boxed{h_j = \sum_{\mu=0}^{m-1} a_\mu f_{j-\mu} - \sum_{\mu=1}^{n} b_\mu h_{j-\mu}\,.} \tag{2.3.61}$$

Es geht also in (2.3.15) über, wenn alle Koeffizienten $b_i = 0$ sind. Ein rekursives Filter hat eine Impulsantwort mit unendlicher Ausdehnung (IIR–Filter). Ihr Vorteil ist, dass die Rechenkomplexität oft deutlich reduziert wird. Zu ihrer Realisierung kann z. B. von einem Filter mit endlicher Impulsantwort g_μ (FIR–Filter) ausgegangen werden und Koeffizienten a_μ, b_μ bestimmt werden, sodass die Impulsantwort des rekursiven Filters möglichst gut mit der des nichtrekursiven übereinstimmt. Die Verfahren dafür sind der angegebenen Literatur zu entnehmen.

Als ein Beispiel für ein rekursives Filter wird hier lediglich eine Approximation des oben in (2.3.41) erwähnten GAUSS-Filters angegeben; die Einzelheiten sind der Literatur zu ent-

nehmen. Da dieses eine separierbare Funktion ist, reicht die Betrachtung des eindimensionalen Falles. Um ein Filter mit dem Parameter σ, der den Durchlassbereich bestimmt, zu realisieren, berechnet man zunächst einen Parameter

$$q = \begin{cases} 0,98711\sigma - 0,96330 & : \quad \sigma > 2,5 \\ 3,97156 - 4,14554\sqrt{1 - 0,26891\sigma} & : \quad 0,5 \leq \sigma \leq 2,5 \end{cases} . \tag{2.3.62}$$

Mit q berechnet man die Filterparameter

$$\begin{aligned} b_0 &= 1,57825 + 2,44413q + 1,4281q^2 + 0,422205q^3 , \\ b_1 &= 2,44413q + 2,85619q^2 + 1,26661q^3 , \\ b_2 &= -\left(1,4281q^2 + 1,26661q^3\right) , \\ b_3 &= 0,422205q^3 , \\ b &= 1 - \frac{1}{b_0}\left(b_1 + b_2 + b_3\right) . \end{aligned} \tag{2.3.63}$$

Die eigentliche Filterung besteht aus einer *Vorwärtsfilterung* in aufsteigenden Indizes, bzw. von links nach rechts, des zu filternden Musters $[f_j]$, die ein Zwischenergebnis $[\phi_j]$ ergibt, gefolgt von einer *Rückwärtsfilterung* in absteigenden Indizes von $[\phi_j]$, bzw. von rechts nach links, die das Endergebnis $[h_j]$ ergibt, nämlich

$$\phi_j = bf_j + \frac{1}{b_0}\left(b_1\phi_{j-1} + b_2\phi_{j-2} + b_3\phi_{j-3}\right) , \tag{2.3.64}$$

$$h_j = b\phi_j + \frac{1}{b_0}\left(b_1h_{j+1} + b_2h_{j+2} + b_3h_{j+3}\right) . \tag{2.3.65}$$

Die Genauigkeit der Approximation ist sehr gut; die Rechenkomplexität ist *nicht* von der Wahl von σ abhängig, im Unterschied zu der nichtrekursiven Form; die Rechenkomplexität ist für $\sigma > 1$ geringer als die der Realisierungen über die nichtrekursive Version, die FFT oder die Kaskadierung von Filtern.

Auf diese Weise lassen sich auch Ableitungen der GAUSS-Funktion rekursiv realisieren. Die *erste Ableitung* erhält man, indem man (2.3.64) *ersetzt* durch

$$\phi_j = \frac{b}{2}\left(f_{j+1} - f_{j-1}\right) + \frac{1}{b_0}\left(b_1\phi_{j-1} + b_2\phi_{j-2} + b_3\phi_{j-3}\right) \tag{2.3.66}$$

und (2.3.65) *unverändert* lässt. Die *zweite Ableitung* erhält man, indem man (2.3.64) *und* (2.3.65) *ersetzt* durch

$$\phi_j = b\left(f_j - f_{j-1}\right) + \frac{1}{b_0}\left(b_1\phi_{j-1} + b_2\phi_{j-2} + b_3\phi_{j-3}\right) , \tag{2.3.67}$$

$$h_j = b\left(\phi_{j+1} - \phi_j\right) + \frac{1}{b_0}\left(b_1h_{j+1} + b_2h_{j+2} + b_3h_{j+3}\right) . \tag{2.3.68}$$

Auch die *anisotrope* Version des GAUSS-Filters, gegeben durch die Gleichungen

$$\begin{aligned} g_{\mathrm{a}}(u,v) &= \frac{1}{2\pi\sigma_u\sigma_v}\exp\left[-\frac{u^2}{2\sigma_u^2} - \frac{v^2}{2\sigma_v^2}\right] \\ \begin{pmatrix} u \\ v \end{pmatrix} &= \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} , \end{aligned} \tag{2.3.69}$$

kann rekursiv realisiert werden, wofür auf die Literatur verwiesen wird.

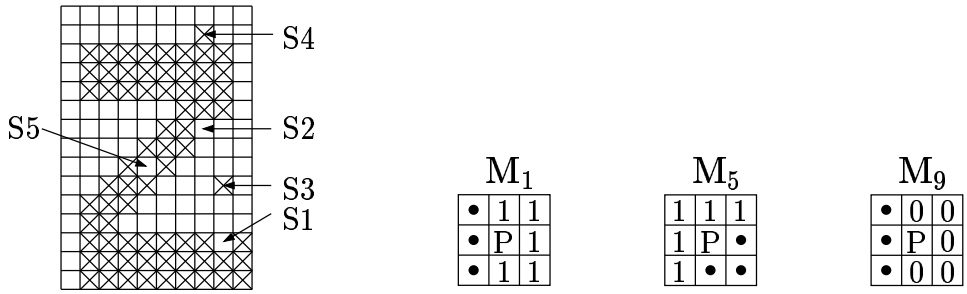


Bild 2.4.1: Zur Beseitigung von Störungen in binären Mustern durch Masken. Die Masken $M_{i+\nu}, \nu = 1, 2, 3$ erhält man, wenn man die Masken $M_i, i = 1, 5, 9$ um $90°, 180°, 270°$ dreht

2.4 Nichtlineare Operationen (VA.1.2.3, 04.12.2005)

2.4.1 Binäre Masken

Wie schon in Abschnitt 2.2.1 erwähnt, werden bei der Klassifikation einfacher Objekte, wie Schriftzeichen oder Werkstücke, oft nur Binärbilder verarbeitet, deren Grauwerte 0 oder 1 sind. Die obigen linearen Operationen in (2.3.15), S. 89, sowie (2.3.40) – (2.3.60), S. 105, liefern jedoch auch bei Anwendung auf binäre Muster i. Allg. keine Ergebnisse $[h_{jk}]$, deren Werte ebenfalls wieder binär sind. Dieses muss dann in einer nachfolgenden Schwellwertoperation (2.2.1), S. 77, durch die eine Nichtlinearität eingeführt wird, sichergestellt werden. Eine andere Möglichkeit besteht darin, für binäre Muster spezialisierte Operationen zur Verbesserung der Qualität zu entwickeln; dafür liegen zahlreiche Beispiele vor. Schließlich haben die linearen Operationen für bestimmte Zwecke Nachteile; z. B. werden bei der Glättung mit (2.3.40) nicht nur hochfrequente Störungen reduziert sondern auch Bildkonturen verschliffen. Aus diesen Gründen wurden **nichtlineare Operationen** entwickelt, von denen im Folgenden einige diskutiert werden.

Bei der Glättung von binären Mustern werden vielfach durch Anwendung von **binären Masken** kleine Störstellen, die beispielsweise nur einen Rasterpunkt groß sind, beseitigt. Die Anwendung einer binären Maske auf ein binäres Muster liefert als Ergebnis wieder ein binäres Muster. Das Prinzip geht aus Bild 2.4.1 hervor. Der Punkt P der Masken $M_i, i = 1, \ldots, 12$ wird auf jeden der Bildpunkte f_{jk} gelegt. Mit $H(M_i)$ werde eine logische Funktion bezeichnet, die den Wert 1 oder „wahr“ annimmt, wenn die Bildpunkte in der 8–Nachbarschaft von f_{jk} die in der Maske angegebenen Werte haben, wobei ein Wert $\bullet$ in der Maske beliebig ist. Einzelne Fehlstellen mit dem Wert 0, wie die Punkte $S1$ und $S2$ in Bild 2.4.1, werden durch die Operation

$$h_{jk} = \begin{cases} 1 & : \quad H(M_i) = 1 \ \text{ für ein } \ i \in \{1, 2, \ldots, 8\} \ , \\ 0 & : \quad \text{sonst} \end{cases} \tag{2.4.1}$$

beseitigt. Einzelne Störpunkte mit dem Wert 1, wie die Punkte $S3$ und $S4$, werden durch die Operation

$$h_{jk} = \begin{cases} 0 & : \quad H(M_i) = 1 \ \text{ für ein } \ i \in \{9, \ldots, 12\} \ , \\ 1 & : \quad \text{sonst} \end{cases} \tag{2.4.2}$$

beseitigt. Mit (2.4.1) werden also „Löcher“ aufgefüllt, mit (2.4.2) wird das Muster von

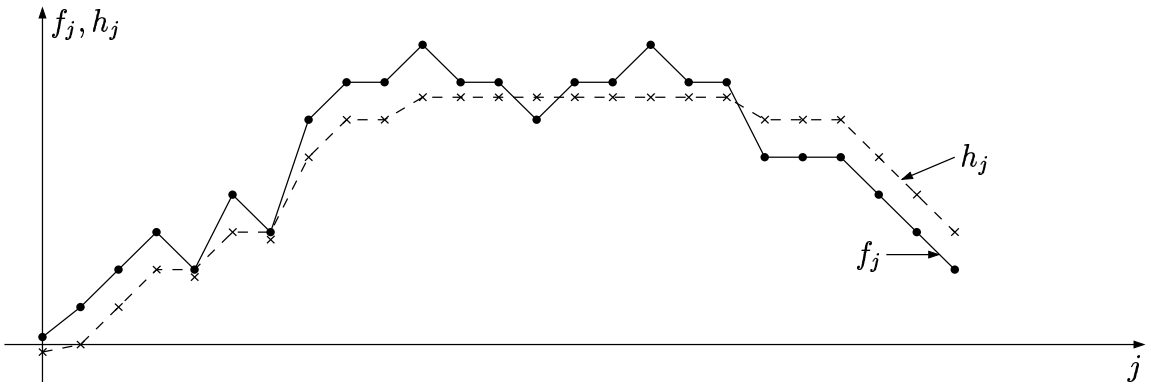


Bild 2.4.2: Ein Beispiel für die Wirkung der nichtlinearen Glättungsoperation (2.4.4). Die Eingangsgröße ist die Folge $[f_j]$ (durchgezogene Linie), das Ergebnis die Folge $[h_j]$ (gestrichelte Linie). Der Schwellwert ist $\theta = 1$.

„Schmutz“ gereinigt. Dagegen bleibt $S5$ mit diesen Masken unverändert. Für spezielle Zwecke kann es erforderlich sein, die Größe der Nachbarschaft oder die Art der Masken zu verändern.

Bezeichnet man die acht Nachbarn des Punktes P der Masken in Bild 2.4.1 mit $f_0^{(P)}, f_1^{(P)}, \ldots, f_7^{(P)}$, z. B. in der durch Bild 2.1.11, S. 76, gegebenen Reihenfolge, so hat die Operation

$$h_{jk} = \begin{cases} 1 & : \ \sum_{j=0}^{7} f_j^{(P)} \geq \theta \ , \\ f_{jk} & : \ \text{sonst} \end{cases} \tag{2.4.3}$$

eine ähnliche Wirkung wie (2.4.1), (2.4.2). Dabei ist θ ein Schwellwert.

Der Funktionsverlauf $f(x)$ eines eindimensionalen Musters oder die Kontur eines zweidimensionalen Musters $f(x, y)$ vor einem Hintergrund lassen sich als Linienmuster mit einer Breite von einem Rasterpunkt auffassen. Der Funktionsverlauf wird wie üblich durch die Folge der diskreten Werte $f_j, \ j = 0, 1, \ldots, M-1$ dargestellt und die Kontur durch die Folge der Koordinatenpaare (x_j, y_j) von den Punkten, die auf der Kontur liegen, wobei die Kontur vom Startpunkt (x_0, y_0) ausgehend z. B. so umlaufen wird, dass das Muster rechts liegt. Eine Glättung derartiger Linienmuster ergibt die nichtlineare Operation

$$\begin{aligned} h_0 &= f_0 \ , \\ h_j &= \begin{cases} h_{j-1} & : \ f_j - \theta \leq h_{j-1} \leq f_j + \theta \ , \\ f_j - \theta & : \ h_{j-1} < f_j - \theta \ , \\ f_j + \theta & : \ h_{j-1} > f_j + \theta \ . \end{cases} \qquad j = 1, 2, \ldots, M-1 \ , \end{aligned} \tag{2.4.4}$$

Eine (zweidimensionale) Konturlinie wird geglättet, indem man (2.4.4) sowohl auf die Folge $[x_j]$ als auch auf $[y_j]$ anwendet, wobei f_j durch x_j bzw. y_j zu ersetzen ist. Ein Beispiel für eine (eindimensionale) Folge $[f_j]$ zeigt Bild 2.4.2. Das Beispiel zeigt, dass nur solche Änderungen von f_j, die größer sind als $\pm\theta$, zu einer Änderung von h_j führen und dass h_j gegen f_j verschoben ist. Letzteres kann durch eine symmetrische Operation vermieden werden. Ähnliche Operationen wie in (2.4.4) wurden auch zur Verbesserung von Schriftzeichen angewendet.

Es sei noch erwähnt, dass die lineare Operation

$$h_1 = f_1 \ ,$$

$$h_j = (1-\alpha)h_{j-1} + \alpha f_j \ , \quad j = 2,3,\ldots,M \tag{2.4.5}$$

eine ähnliche Wirkung wie (2.4.3) hat. Wenn Linienmuster im Kettenkode dargestellt werden, ist es möglich, Glättungsoperationen direkt auf diesem auszuführen.

2.4.2 Rangordnungsoperationen

Das Prinzip der Rangordnungsoperationen ist die Definition einer geeigneten Funktion auf der Rangordnung der Grauwerte in einer Nachbarschaft eines aktuellen Bildpunktes. Wir betrachten einen Funktionswert f_{jk} in der Folge $[f_{jk}]$ und bezeichnen mit N_M eine Nachbarschaft von f_{jk}, die M Werte enthält, z. B.

$$\begin{aligned} N_M &= \{f_{j+\mu,k+\nu} \mid \mu = 0,\pm1,\ldots,\pm m \ ; \ \nu = 0,\pm1,\ldots,\pm n\} \ , \\ M &= (2m+1)(2n+1) \ . \end{aligned} \tag{2.4.6}$$

Die Elemente von N_M werden der Größe nach geordnet, wobei der kleinste Wert aus N_M mit r_1 bezeichnet wird, der nächstgrößere mit r_2 und so weiter. Dieses ergibt die *Rangordnung* der um f_{jk} liegenden Funktionswerte

$$R_{jk} = \{r_1, r_2, \ldots, r_M | r_l \in N_M, \ r_l \le r_{l+1}, \ l = 1,\ldots,M\} \ . \tag{2.4.7}$$

Definition 2.7 *Eine* **Rangordnungsoperation** *ist definiert durch irgendeine Funktion der Rangordnung*

$$h_{jk} = \varphi(R_{jk}) \ . \tag{2.4.8}$$

Zur Definition einer Rangordnungsoperation gehört also die Definition einer geeigneten Nachbarschaft N_M und einer geeigneten Funktion φ, wofür keine theoretisch fundierten Ansätze vorliegen. Beispiele für spezielle und für die Vorverarbeitung nützliche Operationen, die übrigens alle *ohne* Multiplikationen auskommen, sind

$$\begin{aligned} h_{ij} &= r_1 \ , && \text{(Erosion)} \\ h_{ij} &= r_M \ , && \text{(Dilatation)} \\ h_{ij} &= r_M - r_1 \ , && \text{(Konturextraktion)} \\ h_{ij} &= r_{(M+1)/2} \ , && \text{(Median)} \\ h_{ij} &= \begin{cases} r_1 & : \ (f_{ij} - r_1 < r_M - f_{ij}) \ , \\ r_M & : \ \text{sonst} \ . \end{cases} && \text{(Kontrastverstärkung)} \end{aligned} \tag{2.4.9}$$

Der **Median** in (2.4.9) bewirkt eine nichtlineare Glättung, die gegenüber (2.3.40) den Vorteil hat, dass kleine Änderungen völlig beseitigt werden und größere Sprünge im Funktionswert nicht verschliffen werden. Für eine Wahrscheinlichkeitsverteilungsfunktion $P(x)$ ist der Median x_m durch die Gleichung

$$P(x_m) = 0,5 \tag{2.4.10}$$

definiert. Entsprechend wird für ein **Medianfilter** der Breite $(2m+1)$ die empirische Verteilungsfunktion $P(f)$ im Punkte j der Folge $[f_j]$ über die Funktionswerte $f_{j+\nu}$, $\nu = 0,\pm1,\ldots,\pm m$ berechnet; diese entspricht dem normierten Histogramm in Abschnitt 2.2.2.

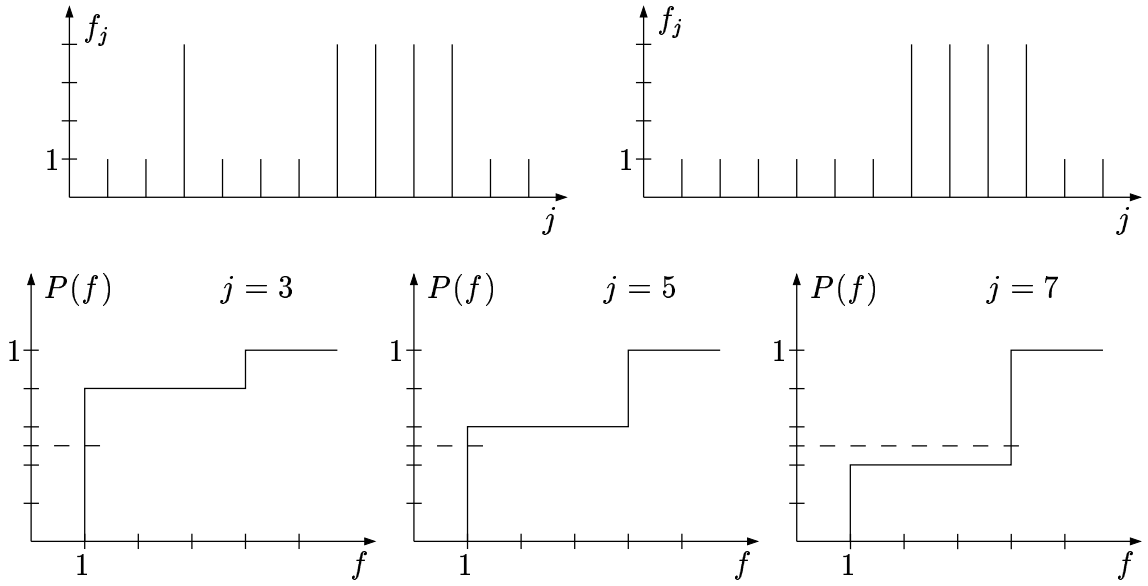


Bild 2.4.3: Eine Folge $[f_j]$ und das Ergebnis $[h_j]$ der Medianfilterung. Die empirische Verteilung $P(f)$ ist dargestellt, wenn die Mitte des Filters der Breite 5 an den Stellen $j = 3, 5, 7$ liegt. Der Einfluss des Randes ist vernachlässigt

Bild 2.4.3 zeigt ein Beispiel für $m = 2$. Wird der Funktionswert $f_{j+\nu} = b$ an q Punkten gemessen, so springt die Funktion $P(f)$ an der Stelle $f = b$ um den Wert $q/(2m + 1)$. Man bestimmt den Wert $f = x_m$, der (2.4.10) genügt, und setzt

$$h_j = x_m \ , \tag{2.4.11}$$

wobei h_j ein Wert der geglätteten Folge $[h_j]$ ist. Das Beispiel zeigt, dass schmale Sprünge im Funktionsverlauf ganz unterdrückt werden und breite Sprünge unverändert bleiben. Die Breite des Medianfilters bestimmt, bis zu welcher Breite ein Sprung noch beseitigt wird. Für zweidimensionale Folgen $[f_{jk}]$ wird analog verfahren.

Für die **Kontrastverstärkung** in (2.4.9) wurde eine Nachbarschaft

$$N_4 = \{f_{jk},\ f_{j-1,k},\ f_{j,k+1},\ f_{j,k-1}\} \tag{2.4.12}$$

vorgeschlagen. Die Operation wird einige Male iteriert bis ein genügend verbessertes Muster vorliegt. Durch die **Erosion** wird ein Muster „verkleinert“, durch die **Dilatation** „vergrößert“. Mit der Operation

$$h_{jk} = r_M - r_1 \tag{2.4.13}$$

werden die Konturen eines Objekts herausgehoben. Ein einfaches Beispiel zeigt Bild 2.4.4.

Ein weiteres Beispiel ist der **kNN–Mittelwert**, der eine kontrasterhaltende Glättung bewirkt und dessen Prinzip Bild 2.4.5 zeigt. Links ist eine Grauwertkante mit einer 3×3 Nachbarschaft angedeutet, rechts die Grauwerte in dieser Nachbarschaft. Der zentrale aktuelle Bildpunkt hat den Grauwert 62. Seine 5 nächsten Nachbarn in der Rangordnung der Grauwerte sind $62,\ 60,\ 55,\ 49,\ 47$. Ihr arithmetischer Mittelwert ist (aufgerundet) 55, und das ist der Wert, der dem aktuellen Bildpunkt als kNN–Mittelwert zugewiesen wird. Der Mittelwert bzw. Median in der 3×3 Nachbarschaft ist 83 bzw. 60.

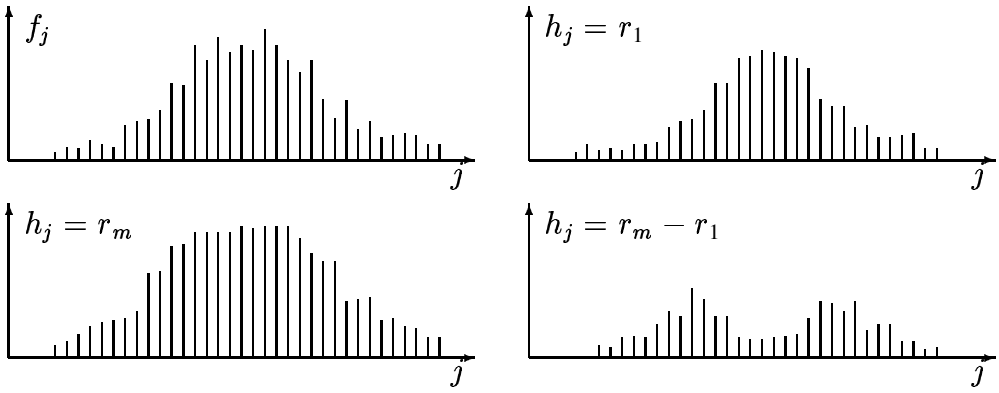


Bild 2.4.4: Zur Wirkung der Rangordnungsoperationen

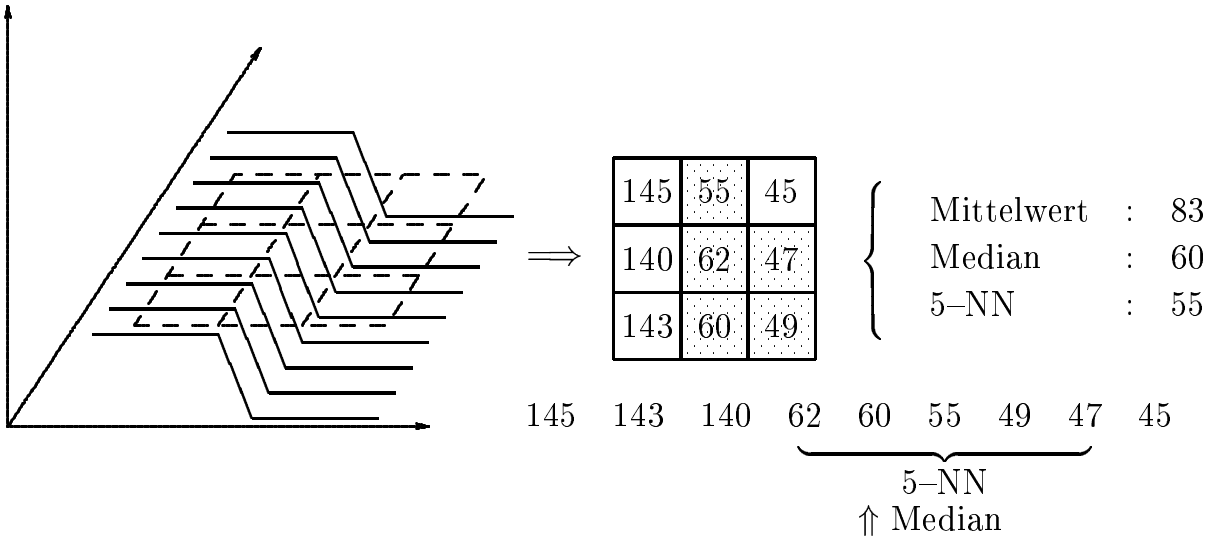


Bild 2.4.5: Prinzip des kNN–Mittelwertes, der über die k Nachbarn (im Grauwert) des aktuellen Bildpunktes gebildet wird

2.4.3 Morphologische Operationen

In (1.2.6), S. 13, wurde ein Muster, alternativ zu (1.2.5), S. 13, durch die Menge der Tupel $\mathbf{f} = \{(\boldsymbol{x},\ \boldsymbol{f_x})\}$ definiert. Mit der in Abschnitt 2.1 beschriebenen Abtastung der Ortsvariablen zu $(x_j = x_0 + j\Delta x, y_k = y_0 + k\Delta y)$ und der Quantisierung der Werte von f zu f_l ergeben sich z. B. für ein Bild die (endlich vielen) Tupel

$$\mathbf{f} = \left\{(j,k,f_{jk})^{\mathsf{T}}\right\}, \quad j = 0,1,\ldots,M_x-1,\ k = 0,1,\ldots,M_y-1,\ f_{jk} \in b_1,\ldots,b_L\ . \tag{2.4.14}$$

Die mathematische **Morphologie** fasst ein Muster als *Menge* von solchen Tupeln auf. Sie definiert Operationen auf Binär–, Grauwert– und Farbbildern als Mengenoperationen. Die Operationen sind insbesondere Vereinigung, Durchschnitt und Komplement, die auf das Muster (die erste Menge) und geeignete Verschiebungen eines Strukturelements (die zweite Menge)

angewendet werden. Es handelt sich um *nichtlineare Nachbarschaftsoperationen*. Das Strukturelement bestimmt die Nachbarschaft. Speziell bei binären Mustern reicht es, die Menge der (diskreten) Koordinatentupel (j,k) anzugeben, für die das Muster den Wert Eins annimmt. Hier wird an der in Abschnitt 2.1 eingeführten Notation festgehalten, die Muster als endliche (ein– oder mehrdimensionale) Folge von Abtastwerten einer Funktion $\boldsymbol{f}(\boldsymbol{x})$ definiert; dafür werden einige elementare Operationen auf Binärmustern angegeben und für weitere auf die Literatur verwiesen. Das **Strukturelement** $\boldsymbol{s}$ ist dann eine Matrix von Koordinatentupeln $\boldsymbol{s} = \{(\mu,\nu)|\mu = -m_-,-m_-+1,\ldots,0,\ldots,m_+-1,m_+;\ \nu = -n_-,n_-+1,\ldots,0,\ldots,n_+-1,n_+\}$. Spezielle Strukturelemente sind solche, die Liniensegmente oder Kreise approximieren sowie Rechtecke, wobei der Referenzpunkt $(0,0)$ des Strukturelements durch Wahl von m_-,n_- festgelegt wird.

Definition 2.8 *Eine elementare* morphologische Operation *für Binärbilder ist die* **Erosion** *mit dem Strukturelement* $\boldsymbol{s}$

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{f} \ominus \boldsymbol{s}\ , \\ h_{jk} &= \begin{cases} 1 & : \quad f_{j+\mu,k+\nu} = 1\ ,\forall\ (\mu,\nu) \in \boldsymbol{s}\ , \\ 0 & : \quad \text{sonst}\ , \end{cases} \\ &= \min_{(\mu,\nu)\in\boldsymbol{s}} \{f_{j+\mu,k+\nu}\}\ . \end{aligned} \tag{2.4.15}$$

Definition 2.9 *Das Gegenstück (aber nicht die Inverse) zur Erosion ist die* **Dilatation** *für Binärbilder mit dem Strukturelement* $\boldsymbol{s}$

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{f} \oplus \boldsymbol{s}\ , \\ h_{jk} &= \begin{cases} 1 & : \quad \exists(\mu,\nu) \in \boldsymbol{s}\ , \quad f_{j+\mu,k+\nu} = 1\ , \\ 0 & : \quad \text{sonst}\ , \end{cases} \\ &= \max_{(\mu,\nu)\in\boldsymbol{s}} \{f_{j+\mu,k+\nu}\}\ . \end{aligned} \tag{2.4.16}$$

Erosion und Dilatation sind in dem Sinne **duale morphologische Operationen** als das Ergebnis der Erosion von $\boldsymbol{f}$ dasselbe ist wie die Dilatation des Komplements $\boldsymbol{f}^c = 1 - \boldsymbol{f}$ und die Verwendung des Komplements der Operation als Ergebnis. Die Operationen werden in Bild 2.4.6 erläutert. Der Ursprung des Strukturelements wird auf den aktuellen Bildpunkt (j,k) gelegt und die Bedingungen (2.4.15) bzw. (2.4.16) geprüft. Im gezeigten Fall ergibt die Erosion $h_{jk} = 0$, d. h. das Binärmuster wird am Rand verkleinert, die Dilatation ergibt $h_{jk} = 1$. Die Bedingung für die Dilatation bleibt auch noch erfüllt, wenn der Punkt (j,k) aus dem Binärmuster herausgezogen wird, das Strukturelement aber noch eintaucht, d. h. das Binärmuster wird am Rand vergrößert.

Die oben für Binärbilder definierten Operationen lassen sich in der Formulierung mit den Minimum– und Maximumoperationen auch auf Grauwertbilder anwenden. Eine Verallgemeinerung ergibt sich durch Verwendung eines Strukturelements $\boldsymbol{s} = \{\mu,\nu,s_{\mu\nu}\}$, das für jedes Koordinatentupel (μ,ν) einen Wert $s_{\mu\nu}$ aus den Zahlen $[s_{\min},s_{\max}]$ annehmen kann. Man entnimmt den Definitionen, dass ein Strukturelement mit M Bildpunkten, die alle den Wert Null haben, genau wieder die Erosion bzw. Dilatation in (2.4.15) bzw. (2.4.16) oder auch in (2.4.9) ergibt. Ansätze zur Bestimmung eines Strukturelementes und einer Folge von morphologischen Operationen werden in der Literatur genannt.

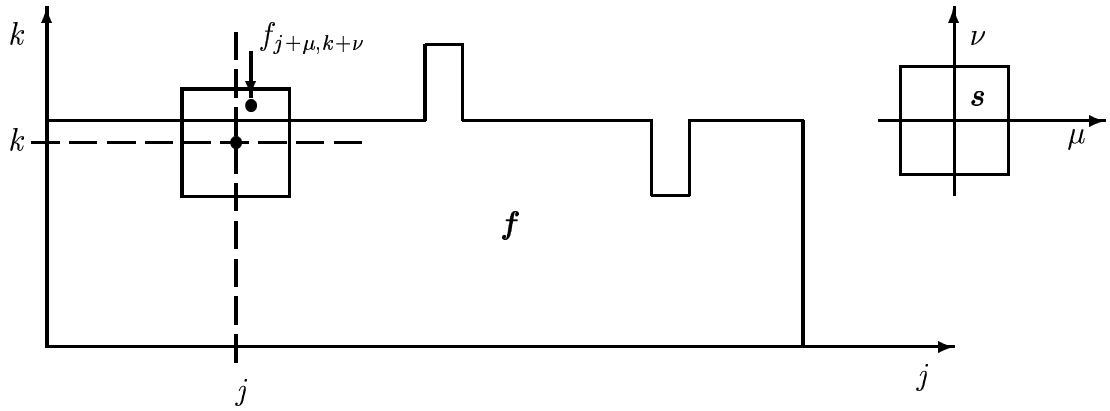


Bild 2.4.6: Einsatz des Strukturelements $\boldsymbol{s}$ bei der Anwendung der morphologischen Operation auf ein Muster $\boldsymbol{f}$

Definition 2.10 *Die* **Erosion für Grauwertbilder** *ist definiert durch*

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{f} \ominus \boldsymbol{s} \;, \\ h_{jk} &= \min_{\mu,\nu \in \boldsymbol{s}} \{f_{j+\mu,k+\nu} - s_{\mu\nu}\} \;. \end{aligned} \tag{2.4.17}$$

Definition 2.11 *Die* **Dilatation für Grauwertbilder** *ist definiert durch*

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{f} \oplus \boldsymbol{s} \;, \\ h_{jk} &= \max_{\mu,\nu \in \boldsymbol{s}} \{f_{j+\mu,k+\nu} + s_{\mu\nu}\} \;. \end{aligned} \tag{2.4.18}$$

Eine Folge von morphologischen Operationen, die u. U. auch iteriert werden, ergibt einen *morphologischen Algorithmus*. Zwei Standardbeispiele für die Verkettung von morphologischen Operationen sind die **Öffnung** (“opening”) und **Schließung** (“closing”)

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{f} \bigcirc \boldsymbol{s} = (\boldsymbol{f} \ominus \boldsymbol{s}) \oplus \boldsymbol{s} \;, \quad \text{(Öffnung)} \;, \\ \boldsymbol{h} &= \boldsymbol{f} \circ \boldsymbol{s} = (\boldsymbol{f} \oplus \boldsymbol{s}) \ominus \boldsymbol{s} \;, \quad \text{(Schließung)} \;, \end{aligned} \tag{2.4.19}$$

wobei wir hier von symmetrischen Strukturelementen ausgehen. Die Wirkung der Operationen auf ein Binärmuster zeigt Bild 2.4.7. Man sieht, dass die Schließung in Bild 2.4.7d) alle Hintergrundelemente, d. h. solche mit *geringer* Helligkeit, füllt, die nicht das Strukturelement aufnehmen können, während die Öffnung in Bild 2.4.7e) alle Objektelemente, d. h. solche mit *großer* Helligkeit, eliminiert, die nicht das Strukturelement aufnehmen können. Die wiederholte Anwendung von Schließungen ändert ein Ergebnis nicht mehr, ebensowenig die wiederholte Anwendung von Öffnungen; die Operationen sind also *idempotent*.

Zur Segmentierung von Bildregionen kann die **Wasserscheidentransformation** genutzt werden. Man betrachtet das Bild $[f_{jk}]$ als Grauwertrelief über der Bildebene, dessen lokale Minima unten durchlöchert werden. Das Relief wird dann langsam in ein Wasserbecken getaucht, sodass Wasser an allen Minima einfließt, an den tiefsten zuerst, dann sukzessive an höher gelegenen lokalen Minima. Dort wo das Wasser aus zwei Minima zusammenfließen würde, werden Dämme errichtet. Am Ende der Flutung ist jedes lokale Minimum von einem Damm umgeben, der die segmentierten Bereiche abgrenzt.

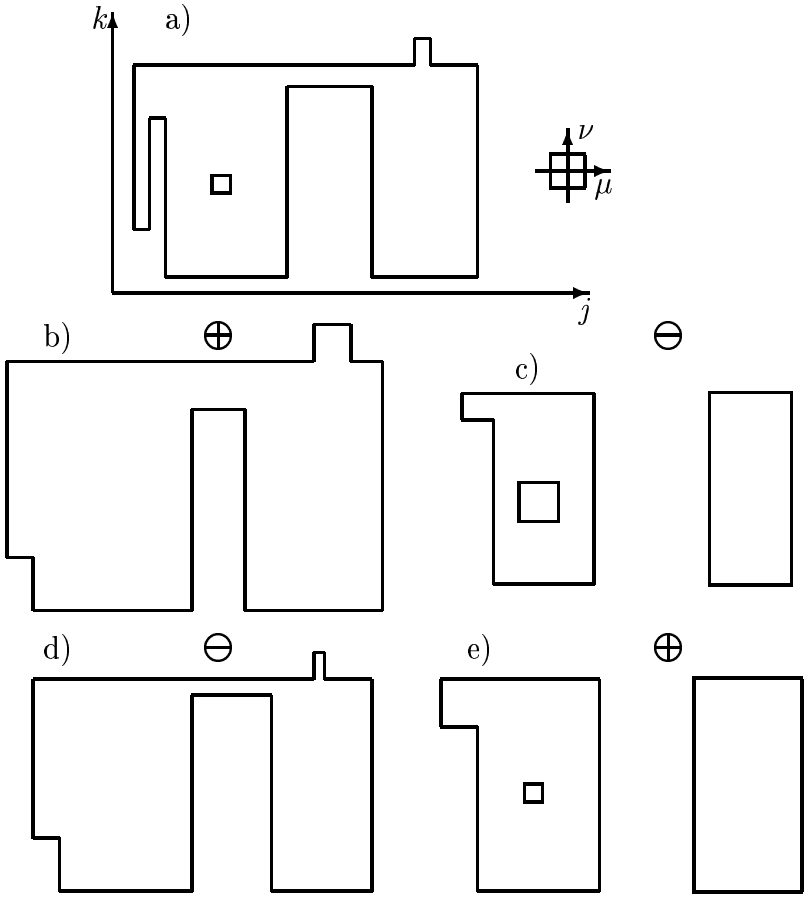


Bild 2.4.7: a) gegebenes Binärbild; die Wirkung der Operationen b) Dilatation bzw. c) Erosion auf das gegebene Bild; die Wirkung der Operationen d) Schließung (Erosion des Bildes in b)) bzw. e) Öffnung (Dilatation des Bildes in c))

Für weitere morphologische Operationen wird auf die angegebene Literatur verwiesen, ebenso zu ihrer effizienten Realisierung.

2.4.4 Diffusionsfilter

Ein wesentlicher Vorteil der Diffusionsfilter ist, dass die Verarbeitung von lokalen Eigenschaften des Musters abhängt, während z. B. die Faltung in Abschnitt 2.3.2 mit einer konstanten Impulsantwort arbeitet und damit in gleicher Weise auf alle Teilintervalle eines Musters angewendet wird. Der Preis ist eine erhöhte Komplexität der Rechnungen.

Ein Diffusionsvorgang wird durch einen Konzentrationsgradienten ∇v in einem Medium v verursacht. Der Gradient verursacht eine Strömung $\boldsymbol{\varphi}$, die dem Gradienten entgegengerichtet ist (z. B. fließt auch Wärme vom heißen zum kalten Bereich), d. h. $\boldsymbol{\varphi} = -\boldsymbol{D}\nabla v$, wobei $\boldsymbol{D}$ der *Diffusionstensor* ist. Sind ∇v und $\boldsymbol{\varphi}$ parallel, so liegt ein *isotroper* Diffusionsvorgang vor und $\boldsymbol{D}$ reduziert sich auf ein skalares *Diffusionsvermögen* $d(x, y)$. Ein Diffusionsvorgang erzeugt bzw. vernichtet nichts von dem Medium sondern transportiert es nur. Eine positive zeitliche Änderung wird daher kompensiert durch eine entsprechende negative Volumenableitung. Dieses wird in der *Kontinuitätsgleichung* $\partial v/\partial t = -\mathrm{div}\boldsymbol{\varphi}$ formal ausgedrückt. Damit erhält man die

Diffusionsgleichung

$$\frac{\partial v(x,y,t)}{\partial t} = \operatorname{div}\left[\boldsymbol{D}\nabla\boldsymbol{v}\right] = \frac{\partial(\boldsymbol{D}\nabla\boldsymbol{v})_x}{\partial x} + \frac{\partial(\boldsymbol{D}\nabla\boldsymbol{v})_y}{\partial y} \; . \tag{2.4.20}$$

Dabei ist $v(x,y,t)$ eine Funktion, die die *Konzentration* eines Mediums an der Stelle (x,y) zur Zeit t angibt, und der Operator div ist die **Divergenz** eines Vektorfeldes. Wenn ein von Null verschiedener Konzentrationsgradient besteht, führt das im Verlauf der Zeit zu einer Konzentrationsänderung, die so gerichtet ist, dass der Konzentrationsgradient reduziert wird. Als Anfangsbedingung wird $v(x,y,0) = f(x,y)$ gesetzt, d. h. die anfängliche „Konzentration“ ist das zu filternde Bild.

Setzt man im einfachsten Falle $\boldsymbol{D} = \boldsymbol{I}$ mit $\boldsymbol{I}$ der Einheitsmatrix, so ergibt sich

$$\begin{aligned} \frac{\partial v}{\partial t} &= \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \; , \\ v(x,y,0) &= f(x,y) \; . \end{aligned} \tag{2.4.21}$$

Es ist bekannt, dass diese partielle Differentialgleichung die Lösung hat

$$v(x,y,t) = \begin{cases} f(x,y) & : \quad t = 0 \; , \\ f(x,y) * \mathcal{N}(x,y;0,\sqrt{2t}) & : \quad t > 0 \; . \end{cases} \tag{2.4.22}$$

Die Lösung dieser speziellen Diffusionsgleichung führt also auf die *Faltung* des gegebenen Musters mit einer GAUSS-Funktion (s. (2.3.41), S. 101), deren Streuung $\sigma = \sqrt{2t}$ mit der Zeit anwächst, d. h. aus Sicht von Abschnitt 2.3 auf nichts Neues.

Interessante und praktisch wichtige Verallgemeinerungen ergeben sich, wenn der Diffusionstensor *nicht* die Einheitsmatrix ist; dieses resultiert in *nichtlinearen Diffusionsfiltern*. Einige mögliche Beispiele sind

$$D_1 = d\left(|\nabla v|^2\right) = \frac{1}{1 + \frac{|\nabla v|^2}{\lambda^2}} \qquad \text{(isotrop)} \; , \tag{2.4.23}$$

$$D_2 = d\left(|\nabla v_\sigma|^2\right) \qquad \text{(regularisiert, isotrop)} \; , \tag{2.4.24}$$

$$\text{mit } v_\sigma = \mathcal{N}(x,y;0,\sigma) * v = \mathcal{N}_\sigma * v \; ,$$

$$D_3 = D\left(\mathcal{N}(x,y;0,\rho) * (\nabla v_\sigma \nabla v_\sigma^{\mathsf{T}})\right) \; , \qquad \text{(anisotrop)} \; . \tag{2.4.25}$$

Hier wird nur der Fall in (2.4.24) genauer betrachtet, für Erweiterungen wird auf die Literatur verwiesen. Die *Regularisierung* des Gradienten durch Faltung mit einer GAUSS-Funktion macht die Lösung robuster gegenüber Störungen. Damit ergibt sich die Diffusionsgleichung

$$\frac{\partial v(x,y,t)}{\partial t} = \frac{\partial}{\partial x}\left(d\left(|\nabla v_\sigma|^2\right)\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(d\left(|\nabla v_\sigma|^2\right)\frac{\partial v}{\partial y}\right) = d_x + d_y \; . \tag{2.4.26}$$

Satz 2.13 *Die Diffusionsgleichung (2.4.26) hat eine eindeutige Lösung, die beliebig oft differenzierbar ist.*
Beweis*: s. z. B. [Catte et al., 1992].*

Die obige Gleichung wird numerisch gelöst und dafür diskretisiert. Der Einfachheit halber wird zunächst nur der Term $\partial/\partial x$ betrachtet. Gesucht ist also eine Diskretisierung von

$$d_x = \frac{\partial}{\partial x}\left(d\left(|\nabla v_\sigma|^2\right)\frac{\partial v}{\partial x}\right) \; . \tag{2.4.27}$$

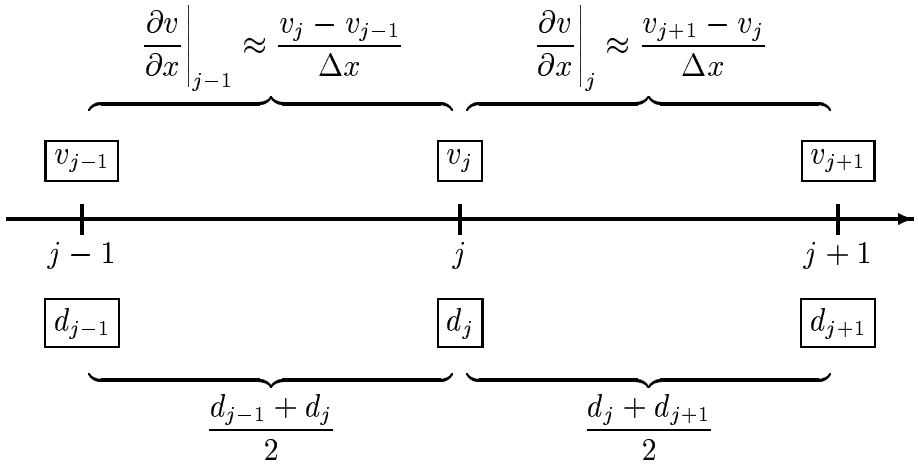


Bild 2.4.8: Zur Diskretisierung des Terms d_x, (2.4.27), in der Diffusionsgleichung (2.4.26)

Zur Abkürzung wird, analog zu (2.1.1), S. 62, gesetzt

$$\begin{aligned} v_{jk\tau} &= v(x_0 + j\Delta x, y_0 + k\Delta y, t_0 + \tau\Delta t) \,, \\ d_{jk\tau} &= d\left(|\nabla v_\sigma|^2\right)\big|_{x_0+j\Delta x, y_0+k\Delta y, t_0+\tau\Delta t} \,. \end{aligned} \tag{2.4.28}$$

Eine einfache Diskretisierung zeigt Bild 2.4.8. Danach ist

$$\begin{aligned} \alpha &= d\frac{\partial v}{\partial x}\bigg|_j \approx \frac{d_{j+1,k\tau} + d_{jk\tau}}{2}\frac{v_{j+1,k\tau} - v_{jk\tau}}{\Delta x} \,, \\ d_x &= \frac{\partial}{\partial x}\left(d\frac{\partial v}{\partial x}\right)\bigg|_j \approx \frac{1}{\Delta x}\left(d\frac{\partial v}{\partial x}\bigg|_j - d\frac{\partial v}{\partial x}\bigg|_{j-1}\right) \,, \\ d_x &\approx \frac{1}{\Delta x}\left(\frac{d_{j+1,k\tau} + d_{jk\tau}}{2}\frac{v_{j+1,k\tau} - v_{jk\tau}}{\Delta x} - \frac{d_{jk\tau} + d_{j-1,k\tau}}{2}\frac{v_{jk\tau} - v_{j-1,k\tau}}{\Delta x}\right) \qquad (2.4.29) \\ &= \frac{1}{2(\Delta x)^2}\Big(v_{j-1,k\tau}(d_{jk\tau} + d_{j-1,k\tau}) - v_{jk\tau}(d_{j-1,k\tau} + 2d_{jk\tau} + d_{j+1,k\tau}) \\ &\quad + v_{j+1,k\tau}(d_{jk\tau} + d_{j+1,k\tau})\Big) \,. \qquad (2.4.30) \end{aligned}$$

Mit dem Term d_y wird analog verfahren. Eine diskrete Approximation von (2.4.24) bzw. (2.4.28) ist

$$\begin{aligned} d_{jk\tau} &\approx d\left(\left(\frac{v_{\sigma;j+1,k\tau} - v_{\sigma;j-1,k\tau}}{\Delta x}\right)^2 + \left(\frac{v_{\sigma;j,k+1,\tau} - v_{\sigma;j,k-1,\tau}}{\Delta y}\right)^2\right) \,, \qquad (2.4.31) \\ v_\sigma &= \mathcal{N}_\sigma * v \,, \end{aligned}$$

wobei $d(.)$ z. B. wie in (2.4.23) gewählt wird. Mit der diskreten Approximation der partiellen Ableitung nach der Zeit ergibt sich schließlich eine diskrete Version von (2.4.26) zu

$$\frac{\partial v}{\partial t} \approx \frac{v_{jk,\tau+1} - v_{jk\tau}}{\Delta t}$$

$$= \frac{1}{\Delta x}\left(\frac{d_{j+1,k\tau}+d_{jk\tau}}{2}\frac{v_{j+1,k\tau}-v_{jk\tau}}{\Delta x} - \frac{d_{jk\tau}+d_{j-1,k\tau}}{2}\frac{v_{jk\tau}-v_{j-1,k\tau}}{\Delta x}\right) \quad (2.4.32)$$
$$+\frac{1}{\Delta y}\left(\frac{d_{j,k+1,\tau}+d_{jk\tau}}{2}\frac{v_{j,k+1,\tau}-v_{jk\tau}}{\Delta y} - \frac{d_{jk\tau}+d_{j,k-1,\tau}}{2}\frac{v_{jk\tau}-v_{j,k-1,\tau}}{\Delta y}\right) .$$

In dieser Form gibt es *nur einen* Term zur Zeit $\tau+1$, d. h. die Gleichung lässt sich unter Verwendung von (2.4.30) nach $v_{jk,\tau+1}$ auflösen und ergibt

$$\begin{aligned} v_{jk,\tau+1} &= v_{jk\tau}\left(1 - \frac{\Delta t}{2(\Delta x)^2}(d_{j-1,k\tau}+2d_{jk\tau}+d_{j+1,k\tau})\right. \\ &\qquad \left. -\frac{\Delta t}{2(\Delta y)^2}(d_{j,k-1,\tau}+2d_{jk\tau}+d_{j,k+1,\tau})\right) \\ &\quad +v_{j-1,k\tau}\frac{\Delta t(d_{jk\tau}+d_{j-1,k\tau})}{2(\Delta x)^2} + v_{j+1,k\tau}\frac{\Delta t(d_{jk\tau}+d_{j+1,k\tau})}{2(\Delta x)^2} \\ &\quad +v_{j,k-1,\tau}\frac{\Delta t(d_{jk\tau}+d_{j,k-1,\tau})}{2(\Delta y)^2} + v_{j,k+1,\tau}\frac{\Delta t(d_{jk\tau}+d_{j,k+1,\tau})}{2(\Delta y)^2} . \end{aligned} \quad (2.4.33)$$

Aus der Gleichung geht hervor, dass man zur Berechnung eines neuen Wertes von v an der Stelle (j,k) zur Zeit $\tau+1$ nur den Wert von v an dieser Stelle sowie den seines linken, rechten, oberen und unteren Nachbarn zur Zeit τ braucht. Diese Werte werden mit Faktoren gewichtet und addiert; es ist also eine *nichtlineare Nachbarschaftsoperationen* mit einer 3×3 Maske. Im Unterschied zu den linearen Filtern in Abschnitt 2.3, insbesondere z. B. (2.3.44), S. 102, oder (2.3.57), S. 104, sind jedoch die Faktoren in der Maske hier *nicht konstant*, sondern vom sich im Verlauf der Iteration ändernden Funktionswert abhängig. Die Summe der Gewichte in der Maske ist Eins. Setzt man $\Delta x = \Delta y = 1$ und sichert wie in (2.4.23) $d \le 1$, dann ist die Operation *stabil*, wenn $\Delta t \le 1/4$ ist.

2.5 Normierungsmaßnahmen (VA.1.2.2, 11.06.2004)

2.5.1 Anliegen

Wie in Kapitel 1 dargelegt, kommt es bei der Klassifikation von Mustern darauf an, alle Muster mit gleicher Bedeutung der gleichen Klasse zuzuordnen. Die Muster können sich dabei in vielfältiger Form unterscheiden, wobei die Unterschiede durch geeignete Parameter beschrieben werden. Zum Beispiel kann die Größe von Buchstaben, die Lautstärke von Geräuschen oder die Dauer eines gesprochenen Wortes in weiten Grenzen schwanken, ohne dass dieses Einfluss auf die Bedeutung hat. Prinzipiell kann man anstreben, *Merkmale* zu finden, die invariant gegenüber derartigen Schwankungen von bestimmten Parametern sind, oder man kann versuchen, einen *Klassifikator* zu entwickeln, der unabhängig von auftretenden Schwankungen Muster mit gleicher Bedeutung immer der gleichen Klasse zuordnet. Erfahrungsgemäß ist es aber bei verschiedenen *Parametern* möglich, auftretende Schwankungen mit relativ geringem Aufwand bereits im Rahmen der Vorverarbeitung auszugleichen oder zu *normieren*. Ist $\beta_\nu({}^\varrho\boldsymbol{f})$ der ν-te Parameter von Mustern ${}^\varrho\boldsymbol{f} \in \Omega$, z. B. die Höhe handgeschriebener Ziffern, so werden die Werte von β_ν zwischen einem kleinsten Wert $\beta_{\nu 0}$ und einem größten Wert $\beta_{\nu 1}$ liegen, d. h. es ist

$$\beta_{\nu 0} \leq \beta_\nu({}^\varrho\boldsymbol{f}) \leq \beta_{\nu 1} \ , \quad \forall\, {}^\varrho\boldsymbol{f} \in \Omega \ . \tag{2.5.1}$$

Eine Normierungsmaßnahme ist eine Transformation T_N, die ein vorverarbeitetes Muster

$${}^\varrho\boldsymbol{h} = T_N\{{}^\varrho\boldsymbol{f}\} \tag{2.5.2}$$

liefert, wobei für den Wertebereich

$$\beta'_{\nu 0} \leq \beta_\nu({}^\varrho\boldsymbol{h}) \leq \beta'_{\nu 1} \tag{2.5.3}$$

der transformierten Muster ${}^\varrho\boldsymbol{h}$ die Bedingung

$$\beta'_{\nu 1} - \beta'_{\nu 0} \ll \beta_{\nu 1} - \beta_{\nu 0} \tag{2.5.4}$$

gilt. Der Idealfall ist $\beta'_{\nu 1} - \beta'_{\nu 0} = 0$, da dann alle Muster ${}^\varrho\boldsymbol{h}$ den gleichen Parameterwert, z. B. die gleiche Größe, haben. Offensichtlich darf man nur solche Parameter normieren, deren Werte *keinen* Einfluss auf die *Bedeutung*, d. h. auf die zugehörige Klasse Ω_κ, haben.

Der Zweck der Normierung ist, dass die Merkmale im Merkmalsraum einen kompakteren Bereich bilden und damit die Klassifikation erleichtert wird, wie es in Bild 2.5.1 angedeutet ist; die Kompaktheitshypothese wurde bereits in Abschnitt 1.3 vorgestellt. Bei einem vorgegebenen Aufwand für den Klassifikator wird man also mit normierten Mustern eine kleinere Fehlerwahrscheinlichkeit erwarten können als mit nicht normierten.

Definition 2.12 *Die* **Normierung** *von Mustern soll den Wertebereich von Parametern, die für die Klassifikation irrelevant sind, reduzieren, um bei gegebenem Aufwand für die Klassifikation eine geringere Fehlerwahrscheinlichkeit zu erreichen.*

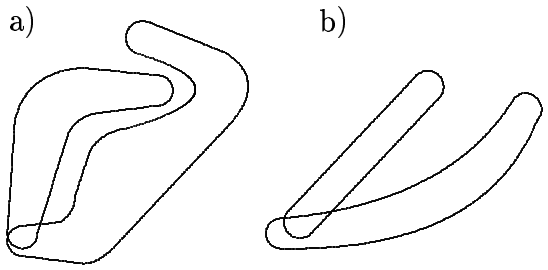


Bild 2.5.1: Merkmale können vor der Normierung komplizierte Bereiche im Merkmalsraum einnehmen a); nach der Normierung sind die Bereiche „kompakter“ geworden, wie z. B. in b)

2.5.2 Interpolation

Ideale Interpolation

Normierungsmaßnahmen erfordern in der Regel eine Transformation T des Musters $f(x, y)$, das nur in Form diskreter Abtastwerte f_{jk} vorliegt. Das Ergebnis sind die Abtastwerte h_{rs} eines transformierten Musters $h(u, v)$. Die Abtastung soll in beiden Fällen an äquidistanten ganzzahligen Stützstellen erfolgen. Die Transformation sei definiert durch

$$\begin{aligned} (x,y)^{\mathsf{T}} &= T\left\{(u,v)^{\mathsf{T}}\right\} \ , \\ h(u,v) &= f(x,y) \ . \end{aligned} \tag{2.5.5}$$

Dabei bewirkt T z. B. eine Skalierung, Translation und/oder Rotation des Musters. Setzt man für (u, v) diskrete ganzzahlige Abtastwerte r, s ein, so erhält man i. Allg. nicht ganzzahlige Werte (x, y), an denen der Funktionswert von f gebraucht wird; dieses ist auch in Bild 2.5.5b gezeigt. Daraus folgt, dass man zwischen den gegebenen Abtastwerten f_{jk} eine **Interpolation** durchführen muss; damit erhält man

$$\boxed{\begin{aligned} h_{rs} &= f(x,y) = f(T\{r,s\}) \ , \\ & \quad r = 0,1,\ldots,M_u-1 \ , \quad s = 0,1,\ldots,M_v-1 \ . \end{aligned}} \tag{2.5.6}$$

Dieser Prozess wird als **Wiederabtastung** (“resampling”) bezeichnte.

Ein üblicher Ansatz für die Interpolation besteht in der gewichteten Summe der Abtastwerte gemäß

$$f(x,y) = \sum_j \sum_k f_{jk}\, g_{\text{int}}(x-j, y-k) \ , \tag{2.5.7}$$

wobei es zunächst offen ist, ob die Summen über j und k endlich sind oder nicht. Dieser Ansatz ist *linear* in den Abtastwerten. Die Gewichtung der Abtastwerte erfolgt durch die Interpolationsfunktion g_{int}. Wir beschränken uns hier auf solche linearen Interpolationsformeln sowie zusätzlich auf *separierbare* Interpolationsfunktionen der Form

$$g_{\text{int}}(x-j, y-k) = g_{\text{int}}(x-j) \cdot g_{\text{int}}(y-k) \ . \tag{2.5.8}$$

Die Verallgemeinerung auf mehr als zwei unabhängige Variable ist offensichtlich. Es reicht also die Betrachtung des eindimensionalen Falles.

Wertet man (2.5.7) für den eindimensionalen Fall an ganzzahligen Werten $x = \mu$ aus, so erhält man die sog. *Interpolationsbedingung*

$$f(\mu) = \sum_j f_j \, g_{\mathrm{int}}(\mu - j) \; . \tag{2.5.9}$$

Für $\mu = j$ sollte sich wieder der Abtastwert f_j ergeben. Das ist dann der Fall, wenn gilt

$$g_{\mathrm{int}}(j) = \begin{cases} 1 & : \quad j = 0 \; , \\ 0 & : \quad j \neq 0 \; , \end{cases} \tag{2.5.10}$$

d. h. wenn die Interpolationsfunktion an allen ganzzahligen Argumenten den Wert Null annimmt, außer beim Argument Null, wo der Wert Eins wird. Ein Vergleich mit (2.3.11), S. 89, zeigt, dass (2.5.9) die *Faltung* von $[f_j]$ und $[g_{\mathrm{int},j}]$ ausgewertet an der Position μ ist.

Aus dem Abtasttheorem ist bekannt, dass die **ideale Interpolation** einer bandbegrenzten Funktion mit (2.1.21) bzw. (2.1.22), S. 65, erfolgt

$$f(x) = \sum_{j=-\infty}^{\infty} f_j \, \frac{\sin[\pi(x-j)]}{\pi(x-j)} = \sum_{j=-\infty}^{\infty} f_j \, \mathrm{sinc}[\pi(x-j)] = \sum_j f_j \, g_{\mathrm{id}}(x-j) \; . \tag{2.5.11}$$

Die Interpolation ist ideal in dem Sinne, dass die abgetastete Funktion $f(x)$ exakt rekonstruiert wird; sie ist aus Sicht der numerischen Berechnung jedoch weniger ideal, da der ideale Interpolator $g_{\mathrm{id}}(x)$ nur langsam abklingt, also viele Werte in der Summe ausgwertet werden müssen, um eine gute Interpolation zu erreichen.

Zur Reduktion der Komplexität werden daher vereinfachte Verfahren verwendet, die einen Kompromiss zwischen Rechenaufwand und Qualität schließen. Für Approximationen mit geringerem Rechenaufwand gibt es zwei Ansätze. Der eine Ansatz besteht darin, dass man Impulsantworten von *endlicher* Ausdehnung sucht, die die ideale möglichst gut approximieren und (2.5.10) genügen. Dafür wurden u. a. unterschiedliche Fensterfunktionen vorgeschlagen, um den idealen Interpolator auf ein endliches Intervall zu beschränken. Der unten betrachtete lineare Interpolator ist der einfachste Ansatz (wenn man von der nächsten Nachbar-Interpolation absieht). Der andere Ansatz besteht darin, von vornherein nur eine auf ein endliches Intervall beschränkte Funktion für die Interpolation zu verwenden und dafür auf die Einhaltung von (2.5.10) zu verzichten. Als Beispiel dafür betrachten wir die Interpolation mit einem B-Spline dritten Grades.

Für den ersten Ansatz gibt es in der in Abschnitt 2.7 zitierten Literatur eine Vielzahl von Ansätzen, wobei auch die verwendeten Qualitätsmaße variieren. Da die FOURIER-Transformierte des idealen Interpolators g_{id} in (2.5.11) eine Rechteckfunktion ist, s. z. B. (2.1.7), S. 64, wird als Qualitätsmaß oft die Abweichung der approximierenden Funktion von der Rechteckfunktion verwendet; auch der subjektive visuelle Eindruck von Interpolationen wird herangezogen.

Lineare Interpolation

Die rechnerisch einfachsten, qualitativ aber auch beschränktesten Verfahren sind die nächster-Nachbar-Interpolation und die lineare Interpolation. Erstere ordnet einem gesuchten Funktionswert $f(x)$ den Wert des nächstliegenden Abtastwertes f_j zu, letztere interpoliert linear zwischen dem rechts und links von $f(x)$ liegenden Abtastwert. Die zugehörigen Impulsantworten g_{nn}

Bild 2.5.2: Lineare Interpolation; (links) einmalige Skalierung um den Faktor 1,73; (mitte) achtmalige Skalierung um den Faktor $\sqrt[8]{2}$ gefolgt von achtmaliger Skalierung um den Faktor $1/\sqrt[8]{2}$; (rechts) elfmalige Rotation um den Winkel $2 \cdot \pi/11$

bzw. g_{lin} sind

$$g_{\mathrm{nn}}(x) = \begin{cases} 0 & : \quad x < -0,5 \\ 1 & : \quad -0,5 \leq x < 0,5 \\ 0 & : \quad 0,5 \leq x \end{cases} , \tag{2.5.12}$$

$$g_{\mathrm{lin}}(x) = \begin{cases} 1 - |x| & : \quad 0 \leq |x| < 1 \\ 0 & : \quad \text{sonst} \end{cases} . \tag{2.5.13}$$

Die nächster-Nachbar-Interpolation führt visuell zu blockartigen Strukturen, die lineare Interpolation verschleift Kontraste. Allerdings sind diese Operationen für Zwecke der Klassifikation, wo ja der visuelle Eindruck nicht im Vordergrung steht, oft schon hinreichend. Beide approximieren den Frequenzgang des idealen Interpolators nur schlecht; sie werden in Abschnitt 2.5.3 im Zusammenhang mit der Normierung der Größe nochmals aufgegriffen.

Bild 2.5.2 zeigt ein Beispiel für die lineare Interpolation des Bildes, dessen Original in Bild 2.2.2, S. 80, oben links gezeigt ist. Man sieht, dass bei nur einmaliger Skalierung der Qualitätsverlust visuell kaum wahrnehmbar ist, bei mehrmaliger Skalierung oder Rotation aber deutlich hervortritt. Auch bei einer einmaligen Rotation ist der Qualitätsverlust kaum wahrnehmbar wie Bild 2.5.4 (links) zeigt. Die Rotationen erfolgten um den Bildmittelpunkt. Zur Vermeidung von Randeffekten wurde das Originalbild spiegelbildlich erweitert, wie es Bild 2.5.4 zeigt, die erweiterte Version rotiert und dann die ursprüngliche Bildgröße dargestellt.

Interpolation mit einem Spline dritten Grades

Bei dem zweiten Ansatz zur Interpolation, den wir zur Unterscheidung hier als *verallgemeinerte Interpolation* bezeichnen, wird von der Gleichung

$$f(x) = \sum_j a_j \, g_{\mathrm{allg}}(x - j) \tag{2.5.14}$$

ausgegangen. Der wesentliche Unterschied zu (2.5.7) besteht darin, dass dort die gewichtete Summe der *Abtastwerte* f_j verwendet wurde, hier jedoch die gewichtete Summe von *Interpolationskoeffizienten* a_j. Damit ergeben sich mehr Freiheitgrade in der Wahl der Interpolations-

funktion, da die Interpolationsbedingung (2.5.9) entfällt. Der Preis ist, dass die Interpolationskoeffizienten einen zusätzlichen Rechenaufwand erfordern, der jedoch bei geeigneter Wahl der verallgemeinerten Interpolationsfunktion g_{allg} gering ist.

Die Interpolationskoeffizienten a_j ergeben sich daraus, dass man (2.5.14) ebenfalls für ganzzahlige Werte $x = \mu$ auswertet

$$f(\mu) = \sum_j a_j \, g_{\text{allg}}(\mu - j) \tag{2.5.15}$$

und fordert, dass die $f(\mu) = f_\mu$ mit den gegebenen Abtastwerten übereinstimmen. Für eine gegebene Funktion g_{allg} ergibt (2.5.15) ein lineares Gleichungssystem zur Bestimmung der a_j. Wenn man Funktionen g_{allg} mit endlichem Definitionsbereich und Funktionen $f(x)$ mit endlich vielen Abtastwerten betrachtet, hat man endlich viele Gleichungen für endlich viele Unbekannte. Es gibt zahlreiche Ansätze zur effizienten Lösung des resultierenden Gleichungssystems, die hier nicht betrachtet werden. Eine Alternative folgt aus der Beobachtung, dass (2.5.15) ebenfalls eine diskrete Faltung ist und dass man folglich die Interpolationskoeffizienten durch Faltung mit der inversen Impulsantwort bzw. der Faltungsinversen berechnen kann, nämlich

$$\begin{aligned} [f_j] &= [a_j] * [g_j] \, , \\ [a_j] &= [g_j]^{-1} * [f_j] \, , \\ \delta_j &= [g_j] * [g_j]^{-1} \, . \end{aligned} \tag{2.5.16}$$

Mit $[g_j]$ wird die diskrete Folge der Werte von $g_{\text{allg}}(j)$ bezeichnet, mit $[g_j]^{-1}$ die Faltungsinverse, deren Existenz zunächst vorausgesetzt wird. Mit δ_j wird der Einheitsimpuls (2.3.3) bezeichnet.

Eine wichtige und sehr leistungsfähige Klasse von verallgemeinerten Interpolationsfunktionen sind die **B-Splines** $\beta^m(x)$ vom Grade $m = 0, 1, \ldots$. Für sie existieren insbesondere effiziente Realisierungen der Faltungsinversen zur Berechnung der Interpolationskoeffizienten. Die B-Splines sind definiert durch

$$\begin{aligned} \beta^0(x) &= \begin{cases} 1 & : \; |x| < 0,5 \\ 0,5 & : \; |x| = 0,5 \\ 0 & : \; |x| > 0,5 \end{cases} \\ \beta^m(x) &= \underbrace{\beta^0(x) * \beta^0(x) * \ldots * \beta^0(x)}_{m \text{ Faltungen}} \, . \end{aligned} \tag{2.5.17}$$

Der B-Spline β^0 stimmt fast mit dem nächster-Nachbar Interpolator g_{nn} überein.

Zwei hier interessierende B-Splines sind der lineare ($m = 1$) und der kubische ($m = 3$) B-Spline, der auch als Spline dritten Grades bezeichnet wird. Der lineare B-Spline ist identisch mit der linearen Interpolationsfunktion g_{lin} in (2.5.13), der kubische ist gegeben durch

$$\beta^3(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{2}{3} & : \; 0 \le |x| < 1 \\ \frac{1}{6}(2 - |x|)^3 & : \; 1 \le |x| < 2 \\ 0 & : \; 2 \le |x| \end{cases} \, . \tag{2.5.18}$$

Sie sind in Bild 2.5.3 gezeigt. Offensichtlich sind sie auf ein endliches und zudem kleines Intervall beschränkt. Das bedeutet, dass es bei Wahl eines B-Splines als verallgemeinerter Interpolationsfunktion g_{allg} in (2.5.14) nur wenige von Null verschiedene Werte und damit wenige Summanden gibt, sodass eine effiziente Berechnung möglich ist.

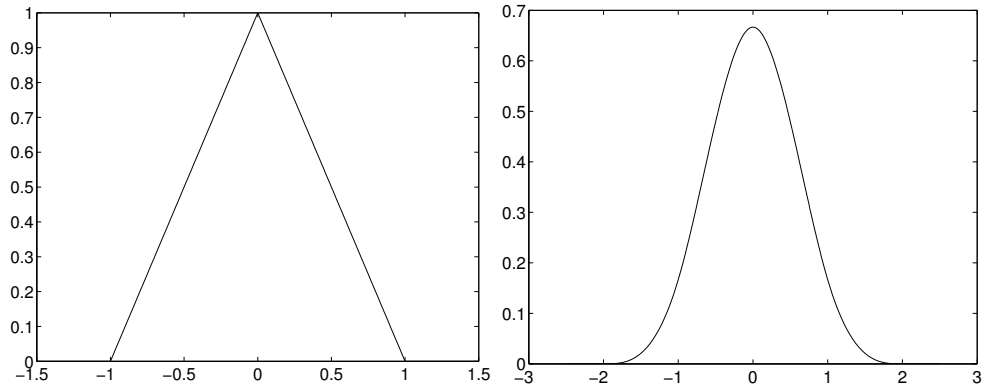


Bild 2.5.3: B-Splines vom Grad $m = 1$ (links) und vom Grad $m = 3$ (rechts)

Die zu klärende Frage ist, wie die Interpolationskoeffizienten a_j für B-Splines berechnet werden. Es zeigt sich, dass für $m = 1$ einfach $a_j = f_j$ gilt, da der B-Spline vom Grade eins eine Interpolationsfunktion $g_{\rm int}$ ist, die auch (2.5.10) genügt. Für $m > 1$ ist die in (2.5.16) erforderliche inverse Operation effizient durch rekursive Filterung realisierbar. Die Rekursionsgleichungen bestehen aus einer Vorwärtsrekurion von links nach rechts mit Koeffizienten a_j^+ und einer Rückwärtsrekursion von rechts nach links mit Koeffizienten a_j^- und lauten für den kubischen B-Spline ($m = 3$)

$$\begin{aligned} z_1 &= \sqrt{3} - 2 \,, \\ a_j^+ &= f_j + z_1 a_{j-1}^+ \,, \quad j = 1, \ldots, M_x - 1 \,, \\ a_j^- &= z_1 \left(a_{j+1}^- - a_j^+ \right) \,, \quad j = M_x - 2, \ldots, 1, 0 \,, \\ a_j &= 6 a_j^- \,. \end{aligned} \tag{2.5.19}$$

Dabei sind, wie üblich, f_j, $j = 0, 1, \ldots, M_x - 1$ die Abtastwerte des Musters. Die Initialisierung erfolgt mit

$$a_0^+ = \sum_{\mu=0}^{\mu_0} f_j z_1^\mu \,, \tag{2.5.20}$$

$$\begin{aligned} \mu_0 &\geq \log[\epsilon] / \log[z_1] \,, \\ a_{M_x-1}^- &= \frac{z_1}{z_1^2 - 1} \left(a_{M_x-1}^+ - z_1 a_{M_x-2}^+ \right) \,. \end{aligned} \tag{2.5.21}$$

Die Variable ϵ ist eine gewünschte Genauigkeit, z. B. $\epsilon = 10^{-6}$. Wenn man ein zweidimensionales Feld von Abtastwerten hat, werden wegen der Separierbarkeit (2.5.8) zunächst alle Zeilen des Feldes mit obigen Gleichungen transformiert, im Ergebnis dann alle Spalten. Das ergibt die zweidimensionalen Interpolationskoeffizienten a_{jk}.

Die Interpolation von nicht ganzzahligen Funktionswerten erfolgt dann mit der Gleichung

$$\boxed{f(x, y) = \sum_{j=j_0}^{j_1} \sum_{k=k_0}^{k_1} a_{jk} \, \beta^m(x - j) \beta^m(y - k) \,.} \tag{2.5.22}$$

Bild 2.5.4: (links) die spiegelbildliche Erweiterung des Eingabebildes nach einmaliger Rotation um den Winkel $2 \cdot \pi/11$ und *linearer* Interpolation; es folgen zwei Bilder mit Interpolation mit einem kubischen Spline; (mitte) achtmalige Skalierung um den Faktor $\sqrt[8]{2}$ gefolgt von achtmaliger Skalierung um den Faktor $1/\sqrt[8]{2}$; (rechts) elfmalige Rotation um den Winkel $2 \cdot \pi/11$

Die Grenzen der Summen werden so gewählt, dass nur Werte, an denen $\beta^m(x-j)$ bzw. $\beta^m(y-k)$ nicht gleich Null ist, erfasst werden

$$\begin{aligned} j_0 &= \left\lceil x - \frac{m+1}{2} \right\rceil , \quad j_1 = j_0 + m , \\ k_0 &= \left\lceil y - \frac{m+1}{2} \right\rceil , \quad k_1 = k_0 + m . \end{aligned}$$

Diese Gleichungen gelten für B-Splines vom Grade m, schließen also auch die lineare Interpolation ein. Die Interpolationskoeffizienten a_{jk} wurden nur für den B-Spline Grad $m = 3$ in (2.5.19) angegeben; für $m = 1$ gilt, wie erwähnt, $a_{jk} = f_{jk}$. Zwei Beispiele für die gute Qualität der Interpolation mit einem kubischen B-Spline zeigen das mittlere und das rechte Bild 2.5.4. Die Gleichungen für ein- bzw. mehrdimensionale Funktionen sind offensichtlich.

2.5.3 Größe

Unabhängig davon, ob ein Muster $\boldsymbol{f}(\boldsymbol{x})$ eine Funktion der Zeit, des Ortes oder sonstiger Variabler ist, wird hier unter der „Größe“ des Musters seine Ausdehnung in den n Koordinatenrichtungen verstanden. Bei einem Buchstaben ${}^{\varrho}f(x,y)$, der im Intervall ${}^{\varrho}x_0 \le x \le {}^{\varrho}x_1$ und ${}^{\varrho}y_0 \le y \le {}^{\varrho}y_1$ liegt, ist z. B. die Größe gegeben durch die Breite ${}^{\varrho}X = {}^{\varrho}x_1 - {}^{\varrho}x_0$ und die Höhe ${}^{\varrho}Y = {}^{\varrho}y_1 - {}^{\varrho}y_0$; bei einem gesprochenen Wort ${}^{\varrho}f(t)$, das zur Zeit ${}^{\varrho}t_0$ beginnt und zur Zeit ${}^{\varrho}t_1$ endet, ist die Größe durch die Dauer ${}^{\varrho}T = {}^{\varrho}t_1 - {}^{\varrho}t_0$ gegeben. Für eine Menge Ω von Mustern ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x})$ gemäß (1.2.5) bedeutet Normierung der Größe, dass das Intervall ${}^{\varrho}x_{\nu 0} \le x_\nu \le {}^{\varrho}x_{\nu 1}$ der ν–ten Koordinate aller Muster ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x}) \in \Omega$ auf ein festes Intervall $x'_{\nu 0} \le x'_\nu \le x'_{\nu 1}$ abgebildet wird. Die Ausdehnung

$${}^{\varrho}X_\nu = {}^{\varrho}x_{\nu 1} - {}^{\varrho}x_{\nu 0} \tag{2.5.23}$$

aller Muster hat dann nach der Normierung (in diesem Falle nach der Skalierung) den festen Wert

$$X'_\nu = x'_{\nu 1} - x'_{\nu 0} \tag{2.5.24}$$

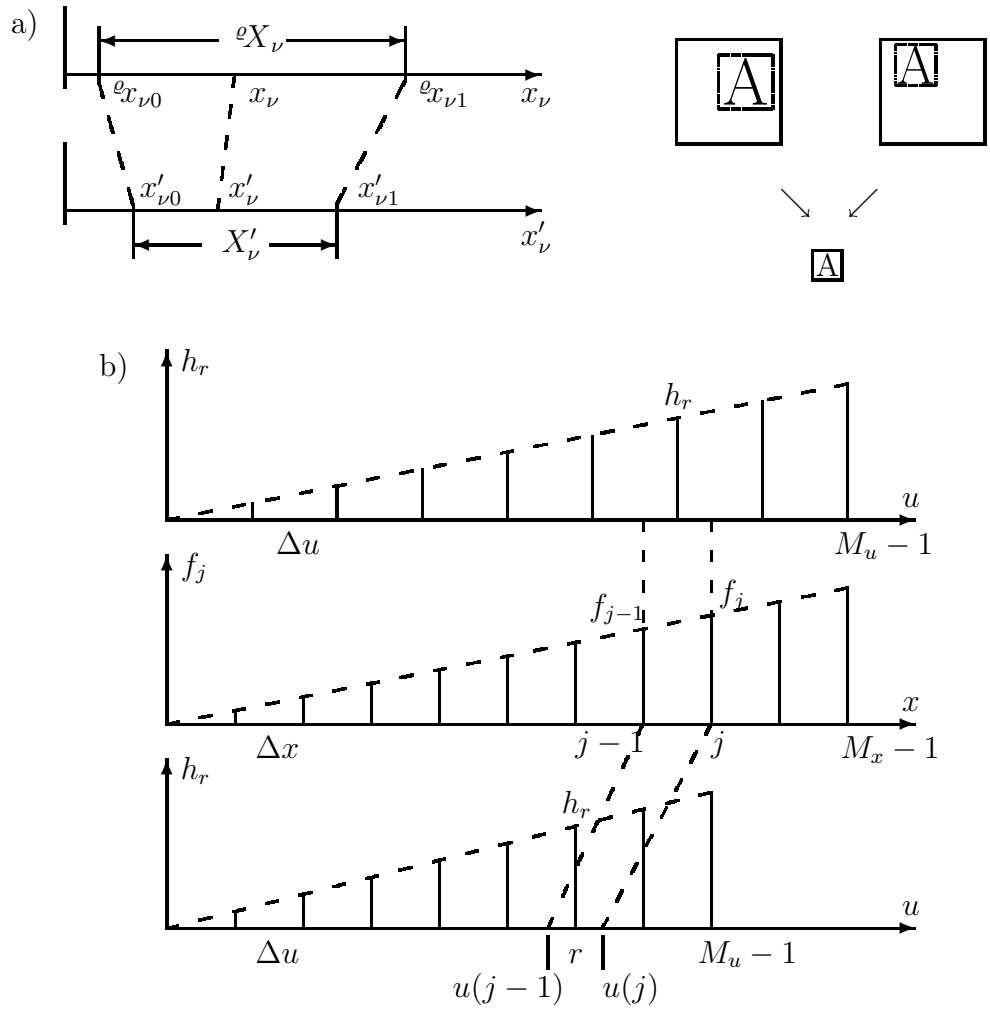


Bild 2.5.5: In a) ist die lineare Abbildung eines Intervalls der Länge $^{\varrho}X_\nu$ auf ein Normintervall der Länge X'_ν angedeutet; Muster wie das „A“ werden dadurch in ein Rechteck fester Größe abgebildet. In b) ist die Normierung einer Folge von Abtastwerten gezeigt

für alle $^{\varrho}\boldsymbol{f}(\boldsymbol{x}) \in \Omega$. Diese Normierung wird für $x_\nu,\ \nu = 1, \ldots, n$ durchgeführt.

Wenn eine lineare Abbildung ausreicht, ist die **Normierung der Größe** relativ einfach. Unter Verwendung der obigen Bezeichnungen gilt, wie aus Bild 2.5.5a hervorgeht, für einen Punkt x_ν und seine Abbildung x'_ν

$$\frac{^{\varrho}x_{\nu 1} - {}^{\varrho}x_{\nu 0}}{x'_{\nu 1} - x'_{\nu 0}} = \frac{x_\nu - {}^{\varrho}x_{\nu 0}}{x'_\nu - x'_{\nu 0}} \,. \tag{2.5.25}$$

Löst man diese Gleichung nach x'_ν auf, so erhält man

$$x'_\nu = x_\nu \frac{X'_\nu}{^{\varrho}X_\nu} + \frac{x'_{\nu 0}\,{}^{\varrho}x_{\nu 1} - {}^{\varrho}x_{\nu 0}x'_{\nu 1}}{^{\varrho}X_\nu} \,. \tag{2.5.26}$$

Ohne Beschränkung der Allgemeinheit kann man jedes Muster so verschieben, dass $^{\varrho}x_{\nu 0} = 0$ ist, und das Normintervall so festlegen, dass $x'_{\nu 0} = 0$ ist. Die etwas umständliche Form (2.5.26) reduziert sich dann auf die einfache Gleichung

$$x'_\nu = x_\nu \frac{X'_\nu}{^{\varrho}X_\nu} \,. \tag{2.5.27}$$

Diese Gleichung bewirkt einfach eine Koordinatentransformation, nämlich eine *Skalierung*, des Musters, die bei kontinuierlichen Mustern im Prinzip problemlos ist.

Bei ihrer Anwendung auf eine eindimensionale Folge $[f_j]$ von *Abtastwerten* erhält man eine Folge h_r von Abtastwerten der normierten Funktion $h(u)$. Wie schon in Abschnitt 2.5.2 erwähnt, ergibt sich das Problem, dass das Bild eines Rasterpunktes auf der x-Achse i. Allg. *zwischen* zwei Rasterpunkten auf der u-Achse liegt. Daher wird dieses Problem, das durch *Wiederabtastung* ("resampling") gelöst wird, noch kurz erörtert; es ist ein Beispiel für eine Transformation gemäß (2.5.5). Zur Vereinfachung der Notation werden die Abtastwerte $[f_j],\ j = 0, 1, \ldots, M_x - 1$ einer Funktion ${}^{\varrho}f(x)$ betrachtet, die im Intervall $x_0 = 0 \leq x \leq M_x - 1 = (M_x - 1)\Delta x = x_1$ liegen, wie es auch in (2.1.2) bereits eingeführt wurde. Dabei ist zu beachten, dass für eine andere Funktion ${}^{\sigma}f(x)$ die Werte M_x und x_1 i. Allg. anders sein werden und für eine mehrdimensionale Funktion auch die weiteren Variablen zu berücksichtigen sind. Die Folge $[f_j]$ soll in eine Folge $[h_r]$ transformiert werden, welche die Abtastwerte derjenigen Funktion $h(u)$ sind, die man durch lineare Abbildung von $f(x)$ in das Intervall $u_0 = 0 \leq u \leq M_u - 1 = (M_u - 1)\Delta u = u_1$ erhält. Mit $\Delta x = \Delta u = 1$ und (2.5.27) erhält man

$$u = j\,\frac{M_u - 1}{M_x - 1} = u(j)\ , \quad \text{bzw.} \quad x = r\,\frac{M_x - 1}{M_u - 1} = x(r) \tag{2.5.28}$$

als Koordinaten der Abtastwerte von $f(x)$ im u-System bzw. von $h(u)$ im x-System. Dieses ist in Bild 2.5.5b unten gezeigt. Nun sind aber die $h_r,\ r = 0, 1, ..., M_u - 1$ Abtastwerte von $h(u)$ an den Stellen $u = r\Delta u = r$. Aus Bild 2.5.5b wird klar, dass i. Allg. der Wert h_r durch eine *Interpolation* der Werte f_j berechnet werden muss. Um den Aufwand für die Interpolation gering zu halten, begnügt man sich oft mit der im Abschnitt 2.5.2 erwähnten *linearen Interpolation*. Es sei $u(j-1)$ der größte Wert mit der Eigenschaft $u(j-1) \leq r$ und $u(j)$ der kleinste Wert mit der Eigenschaft $r \leq u(j)$. Dann erhält man den Abtastwert h_r aus der linearen Interpolationsgleichung

$$h_r = (f_j - f_{j-1})\,\frac{r - u(j-1)}{u(j) - u(j-1)} + f_{j-1}\ , \tag{2.5.29}$$

wobei $u(j) - u(j-1) = (M_u - 1)/(M_x - 1) = \text{const}$ ist. Damit lassen sich Folgen mit variabler Anzahl von Abtastwerten in Folgen mit genau M_u Werten transformieren.

Aus dem oberen Teil von Bild 2.5.5b geht hervor, dass man die Normierung auch als erneute Abtastung von $f(x)$ auffassen kann. Aus $[f_j]$ gewinnt man durch Interpolation zunächst $f(x)$ und tastet diese mit der Schrittweite $\Delta u = x_1/(M_u - 1)$ ab. Ist $M_u < M_x$, so ist $\Delta u > \Delta x$, und aus dem Abtasttheorem in Satz 2.1, S. 65, folgt, dass gegebenenfalls die interpolierte Funktion $f(x)$ *erneut* tiefpassgefiltert werden muss, um die richtige Bandbegrenzung zu erreichen. Auch hier kann man sich auf lineare Interpolation beschränken und entnimmt dafür Bild 2.5.5b oben

$$h_r = (f_j - f_{j-1})\left(r\frac{M_x - 1}{M_u - 1} - j + 1\right) + f_{j-1}\ . \tag{2.5.30}$$

Man überzeugt sich leicht, dass (2.5.29) und (2.5.30) identisch sind. Die Vorgehensweise in Abschnitt 2.5.2 besteht darin, zu einem diskreten Wert $u = r$ von $h(u)$ mit (2.5.5) den zugehörigen x-Wert zu bestimmen, dann mit (2.5.22) den Funktionswert $f(x)$ zu interpolieren und mit (2.5.6) $h_r = f(x)$ zu setzen; dieses wird für $r = 0, 1, \ldots, M_u - 1$ durchgeführt. Offenbar sind alle Vorgehensweisen äquivalent, letztere eignet sich besonders für eine effiziente Ausführung.

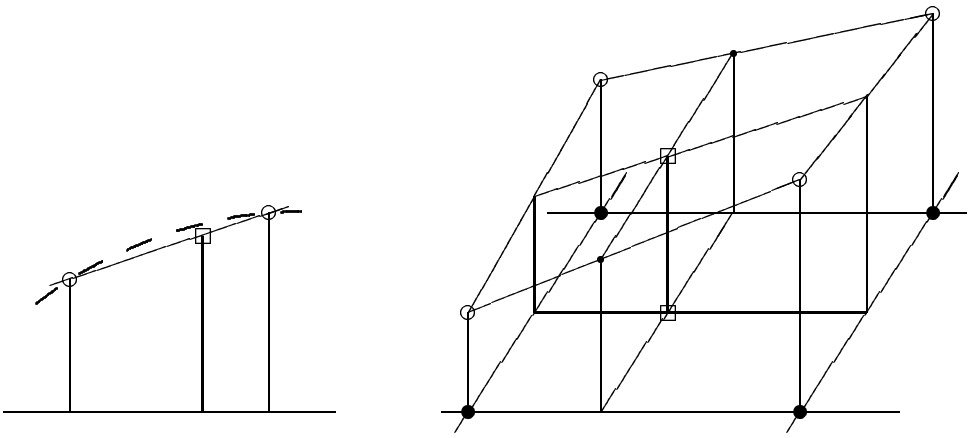


Bild 2.5.6: Schema der bilinearen Interpolation durch wiederholte lineare Interpolation

Bei mehrdimensionalen Funktionen kann man durch mehrfache Anwendung von (2.5.28) auf die n Koordinaten ebenfalls die Rasterpunkte im $\boldsymbol{x}$-Koordinatensystem in das $\boldsymbol{u}$-System abbilden. Auch hier werden i. Allg. die Bilder der Rasterpunkte im $\boldsymbol{x}$-System zwischen denen im $\boldsymbol{u}$-System liegen. Folgende vereinfachte Methoden finden Anwendung. Als Funktionswert eines Rasterpunktes im $\boldsymbol{u}$-System wird der des nächstliegenden Punktes im $\boldsymbol{x}$-System verwendet, die im Abschnitt 2.5.2 erwähnte „nächster-Nachbar-Interpolation“. So würde z. B. nach dieser Methode in Bild 2.5.5b $h_r = f_{j-1}$ gesetzt werden, da der Punkt $u = u(j-1)$ dem Punkt $u = r$ am nächsten liegt. Eine andere Möglichkeit besteht in der Verwendung des Mittelwertes der umliegenden Rasterpunkte. Danach würde man $h_r = (f_{j-1} + f_j)/2$ setzen. Schließlich kann man bei Bildern den Funktionswert durch eine *bilineare Interpolation* gewinnen, wie in Bild 2.5.6 dargestellt. Es wird dabei zunächst z. B. in x-Richtung linear interpoliert, danach zwischen den interpolierten Werten nochmals in y-Richtung. Das gleiche Ergebnis erhält man, wenn man zuerst in y- und dann in x-Richtung interpoliert. Mit (2.5.22) und B-Splines erster Ordnung wird dieser Fall erfasst.

Bei der Normierung von Schriftzeichen oder Werkstücken bestimmt man das kleinste umschreibende Rechteck und bildet dieses linear auf ein Normrechteck ab; bei Wörtern kann die Bestimmung des Anfangs- und Endpunktes Probleme bereiten. Außer den linearen Abbildungen (2.5.26), (2.5.27) sind auch nichtlineare Normierungen möglich. Solche wurden insbesondere im Zusammenhang mit der Worterkennung entwickelt. Sie erfordern jedoch die Kenntnis von Referenzwörtern (Prototypen), und deshalb wird die Diskussion dieser Verfahren auf Kapitel 4 verschoben. Auch (2.5.33) unten bewirkt eine Größennormierung, jedoch nicht auf ein Normintervall sondern auf normierte Momente.

2.5.4 Lage

Die Bedeutung von Mustern ist in vielen Fällen auch unabhängig von einer Translation, manchmal auch unabhängig von einer Rotation. Diesem wird durch eine **Normierung der Lage** Rechnung getragen.

Die Translation ist mit (2.5.26) ebenfalls erfasst, da dadurch ein Intervall der x–Achse sowohl skaliert als auch verschoben wird. Damit wird, wie erwähnt, der Anfangs– und Endpunkt des Musters auf definierte Punkte verschoben. Eine Alternative ist die Verschiebung des Musterschwerpunktes in einen definierten Punkt. Durch die Verwendung von Momenten ist auch

eine Rotation des Musters in eine Normorientierung möglich. Dafür wird eine Folge von Normierungsschritten angegeben. Um die Zahl der Symbole zu reduzieren, wird in jedem Schritt das anfängliche Muster mit $f(x,y)$ bezeichnet, das transformierte mit $h(u,v)$. Das anfängliche Muster im Schritt j ist das transformierte Muster des Schrittes $(j-1)$. Zur Vereinfachung wird mit den kontinuierlichen Formen $f(x,y), h(u,v)$ gearbeitet, da die bereits diskutierten Interpolationsprobleme bei Folgen von Abtastwerten nicht nochmals erörtert werden sollen.

Das gegebene Muster wird in folgenden Schritten normiert, wobei die **Momente** m_{pq} und μ_{pq} der Muster $f(x,y) \geq 0$ und $h(u,v) \geq 0$ definiert sind durch

$$\begin{aligned} m_{pq} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,\mathrm{d}x\,\mathrm{d}y\ , \\ \mu_{pq} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u^p v^q h(u,v)\,\mathrm{d}u\,\mathrm{d}v\ . \end{aligned} \tag{2.5.31}$$

Schritt 1: Mit $x_s = m_{10}/m_{00}$ und $y_s = m_{01}/m_{00}$ setze man

$$\begin{aligned} u &= x - x_s\ , \quad v = y - y_s\ , \\ h(u,v) &= \frac{f(x,y)}{m_{00}}\ . \end{aligned} \tag{2.5.32}$$

Dann ist $\mu_{00} = 1$, $\mu_{10} = \mu_{01} = 0$. Diese auf den Schwerpunkt (x_s, y_s) bezogenen Momente werden auch als **Zentralmomente** bezeichnet.

Schritt 2: Ersetze $f(x,y)$ durch $h(u,v)$, d. h. im Folgenden ist $f(x,y)$ ein Muster mit $m_{00} = 1, m_{10} = m_{01} = 0$. Man setze

$$\begin{aligned} R &= \sqrt{m_{02} + m_{20}}\ , \quad u = \frac{x}{R}\ , \quad v = \frac{y}{R}\ , \\ h(u,v) &= R^2 f(x,y)\ . \end{aligned} \tag{2.5.33}$$

Dann ist $\mu_{20} + \mu_{02} = 1$.

Schritt 3: Ersetze $f(x,y)$ durch $h(u,v)$, d. h. im Folgenden gilt für die Momente von $f(x,y)$ zusätzlich $m_{20} + m_{02} = 1$. Bestimme die Lösungen der Gleichung

$$\tan(2\alpha) = \frac{2m_{11}}{m_{20} - m_{02}}\ . \tag{2.5.34}$$

Führe die Koordinatentransformation (Rotation um den Winkel α)

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{2.5.35}$$

aus, die eine Transformation auf *Hauptträgheitsachsen* ist, und setze

$$h(u,v) = f(x,y)\ . \tag{2.5.36}$$

Dann ist $\mu_{11} = 0$. Von den vier Lösungen von (2.5.34) wähle diejenige, für die $\mu_{20} < \mu_{02}$ und $\mu_{21} > 0$ ist.

Schritt 4: Ersetze $f(x,y)$ durch $h(u,v)$, d. h. für $f(x,y)$ ist zusätzlich $m_{11} = 0$, $m_{20} < m_{02}$, $m_{21} > 0$. Wähle $\beta \in \{+1, -1\}$ so, dass für

$$\begin{aligned} u &= \beta x\ , \quad v = y\ , \\ h(u,v) &= f(x,y) \end{aligned} \tag{2.5.37}$$

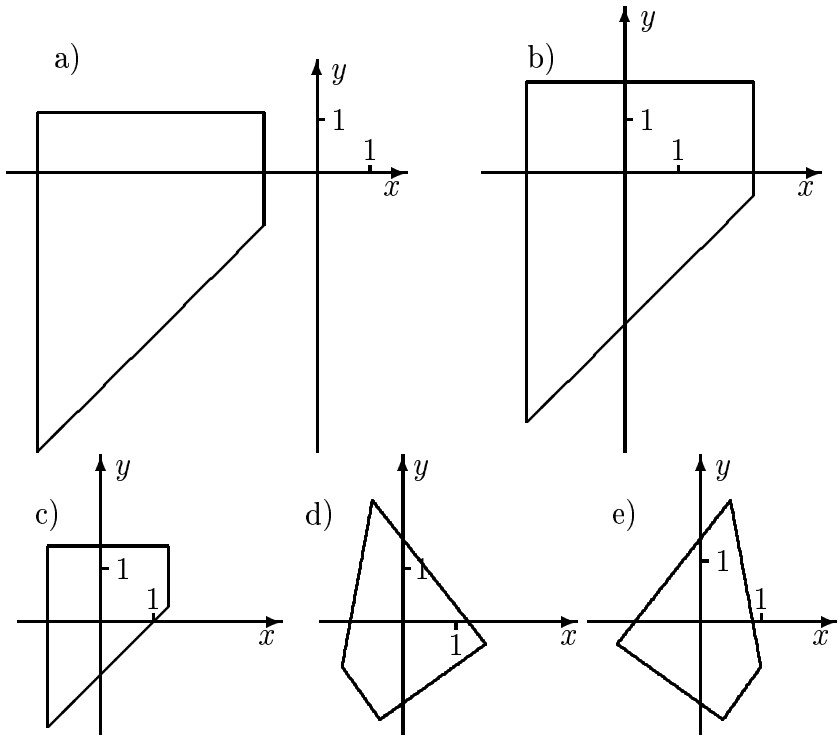


Bild 2.5.7: Normierung eines Musters mit den Gleichungen (2.5.32) – (2.5.37). a) Anfänglich gegebenes Muster, b) Verschiebung des Koordinatensystems in den Schwerpunkt, c) Skalierung der Koordinaten auf $R = 1$, d) Rotation des Musters auf Hauptträgheitsachsen, e) Spiegelung des Musters an der y–Achse

das Moment $\mu_{12} > 0$ ist. Das mit dem letzten Schritt in (2.5.37) erhaltene Muster $h(u, v)$ ist so normiert, dass für seine Momente

$$\begin{aligned} \mu_{00} &= 1 \,, \\ \mu_{10} &= \mu_{01} = \mu_{11} = 0 \,, \\ \mu_{20} + \mu_{02} &= 1 \,, \quad \mu_{20} < \mu_{02} \,, \quad \mu_{21} > 0 \,, \quad \mu_{12} > 0 \end{aligned} \tag{2.5.38}$$

gilt. Damit ist die Normierung abgeschlossen. Sie umfasst eine Translation des Musters und Veränderung der Funktionswerte (2.5.32), eine Skalierung der Koordinaten (2.5.33), eine Rotation (2.5.35) und eine Spiegelung (2.5.37). Bei Bedarf können einige der Normierungsschritte ausgelassen werden. Ein Beispiel für die Wirkung der Normierung gibt Bild 2.5.7. Die Normierung der Drehlage ist insbesondere bei Mustern mit Symmetrieachsen nützlich. Auf die Verwendung von Momenten als Merkmale wird noch in Abschnitt 3.2 eingegangen.

Die hier für ein zweidimensionales Bild angegebenen Beziehungen für Momente lassen sich auf dreidimensionale Objekte übertragen. Die Momente m_{pqr} und Zentralmomente μ_{pqr} eines dreidimensionalen Objekts sind analog (2.5.31) definiert. Eine Normierung auf den Schwerpunkt erfolgt analog zu (2.5.32). Die Normierung der Drehlage kann auf die Hauptträgheitsach-

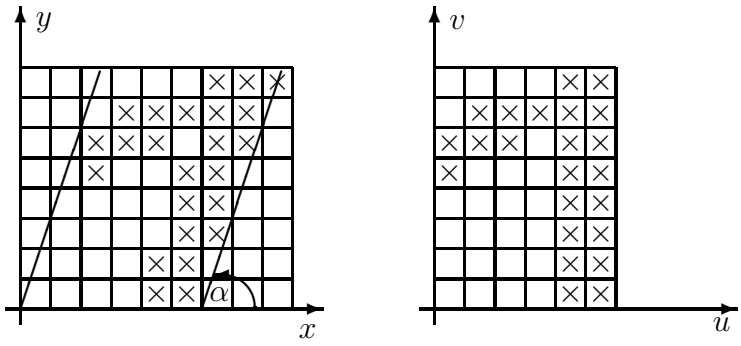


Bild 2.5.8: Aufrichtung einer schrägen Ziffer

sen erfolgen, die durch die Eigenvektoren der Matrix

$$\boldsymbol{M} = \begin{pmatrix} \mu_{200} & \mu_{110} & \mu_{101} \\ \mu_{110} & \mu_{020} & \mu_{011} \\ \mu_{101} & \mu_{011} & \mu_{002} \end{pmatrix} \tag{2.5.39}$$

gegeben sind.

Speziell für die Klassifikation handgedruckter Schriftzeichen entwickelte Lagenormierungen haben auch den Ausgleich der Neigung von Buchstaben zum Ziel, dagegen ist die Rotation von Buchstaben nicht zweckmäßig. Der Einfachheit halber sei angenommen, dass um das geneigte Schriftzeichen ein Parallelogramm gelegt wird. Mit den Bezeichnungen von Bild 2.5.8 wird dieses mit den Gleichungen

$$u = x - y \cot \alpha \ , \quad v = y \tag{2.5.40}$$

in ein Rechteck abgebildet und damit die Schrift aufgerichtet.

2.5.5 Energie

Da die Bedeutung von Mustern häufig auch unabhängig von der im Muster enthaltenen „Energie“ – z. B. der Schallenergie oder der Helligkeit – ist, empfiehlt es sich, auch eine **Normierung der Energie** vorzunehmen, indem man die Amplitude oder die Funktionswerte $f(x, y)$ bzw. f_{jk} normiert. Zunächst wird daran erinnert, dass auch die Schwellwertoperation (2.2.1) eine Normierung der Funktionswerte bewirkt und dass mit (2.5.32) das Integral über das Muster auf den Wert 1 normiert wird.

Die Energie von M zurückliegenden Abtastwerten eines Sprachsignals zum Abtastzeitpunkt $t = j\Delta t = j$ wird mit

$$A_j = \sum_{\nu=0}^{M-1} |\alpha_\nu f_{j-\nu}| \tag{2.5.41}$$

oder auch mit

$$A_j = \sum_{\nu=0}^{M-1} \alpha_\nu f_{j-\nu}^2 \tag{2.5.42}$$

definiert. Dabei ist $\alpha_\nu, \nu = 0, 1, \ldots, M-1$ eine **Fensterfunktion** zur Ausblendung und Gewichtung der Funktionswerte, z. B.

$$\begin{aligned} w_\nu &= 1\ , \quad \nu = 0, 1, \ldots, M-1 && \text{(Rechteckfenster)} && , \\ w_\nu &= 0,54 - 0,46 \cos\left[\frac{2\pi\nu}{M-1}\right] && \text{(HAMMING-Fenster)} && , \\ w_\nu &= 0,5\left(1 - \cos\left[\frac{2\pi\nu}{M-1}\right]\right) && \text{(HANNING-Fenster)} && . \end{aligned} \tag{2.5.43}$$

Neben der Ausblendung von Funktionswerten haben das HAMMING- und das HANNING-Fenster auch den Effekt, dass Abtastwerte in der Nähe von $j = 0$ bzw. $j = M-1$ sehr stark gedämpft werden. Wenn man eine diskrete FOURIER-Transformation durchführt, werden wie in Abschnitt 2.3.3 erörtert, die Abtastwerte periodisch fortgesetzt. Das führt i. Allg. an den Periodengrenzen zu Unstetigkeiten wie auch in Bild 3.2.5, S. 177, deren Effekt durch die genannten Fenster gemildert wird. Mit der Normierung

$$h_k = \frac{f_k}{A_j}\ , \qquad k = j - M + 1, j - M + 2,\ \ldots, j \tag{2.5.44}$$

erreicht man, dass die Energie der Folge $[h_k], k = j - M + 1, \ldots, j$ wegen (2.5.41) den Wert 1 hat. Diese Vorgehensweise lässt sich unmittelbar auf die Abtastwerte mehrdimensionaler Folgen verallgemeinern. Bei (2.5.42) ist (2.5.44) zu modifizieren in

$$h_k = \frac{f_k}{\sqrt{A_j}}\ . \tag{2.5.45}$$

Da man in der Regel mit den Folgen $[f_j]$ oder $[h_k]$ weitere Parameter, insbesondere Merkmale c_ν, berechnen wird, ist es u. U. zweckmäßig, die Merkmale mit den nichtnormierten f_j zu berechnen und nachträglich zu normieren. Damit spart man die vielen Divisionen in (2.5.44), (2.5.45). Die Zahl M der Abtastwerte, über die normiert wird, wird empirisch so ermittelt, dass die Fehlerrate bei der Klassifikation minimiert wird. Ein Erfahrungswert aus der Spracherkennung ist z. B. , dass M etwa gleich der Zahl der Abtastwerte je mittlere Wortlänge ($\approx 0,4$ sec) sein sollte.

Definiert man wie üblich Mittelwert und Streuung über M Abtastwerte einer Folge $[f_j]$ mit

$$\begin{aligned} m &= \frac{1}{M}\sum_{j=0}^{M-1} f_j\ , \\ \sigma &= \sqrt{\frac{1}{M}\left(\sum_{j=0}^{M-1} f_j^2\right) - m^2}\ , \end{aligned} \tag{2.5.46}$$

so erhält man auf Mittelwert 0 und Streuung 1 normierte Werte aus

$$h_j = \frac{f_j - m}{\sigma}\ . \tag{2.5.47}$$

Die Werte h_j sind *invariant* gegenüber einer linearen Transformation der f_j gemäß $af_j + b, \quad a \neq 0$. Sie sind weiterhin *dimensionslos*, was insbesondere dann sinnvoll ist, wenn man Merkmale mit unterschiedlicher Dimension, wie z. B. Länge in m, Gewicht in kg, Preis in DM,

verwendet. Abstandsmaße für einen aus nicht dimensionslosen Komponenten zusammengesetzten Merkmalsvektor sind offenbar nichtssagend. Es sei daran erinnert, dass man eine Normierung der Funktionswerte bereits über die Quantisierungskennlinie eines PCM-Verfahrens erreichen kann, wenn man das Intervall $(f_{\min}, f_{\max})$ in Bild 2.1.7 stets auf die Werte $\{b_1, \ldots, b_L\}$ abbildet. Dabei wird allerdings nicht die Normierung auf die Energie (2.5.41) realisiert. Mit der Gleichung

$$h_j = \alpha \frac{f_j - f_{\min}}{f_{\max} - f_{\min}} , \tag{2.5.48}$$

$$f_{\min} = \min_j \{f_j\} \quad \text{und} \quad f_{\max} = \max_j \{f_j\} \tag{2.5.49}$$

schließlich werden die Werte einer Folge $[f_j]$ auf das Intervall $0 \leq h_j \leq \alpha$ normiert.

Die *Beleuchtung* eines Bildes kann oft inhomogen sein und von einem Bildrand zum anderen von relativ hell zu relativ dunkel variieren. Wenn dieses Bild relativ kleine Objekte (klein im Vergleich zur Bildgröße) enthält, können diese zunächst mit einer morphologischen Schließung beseitigt werden, d. h. es bleibt in etwa nur die inhomogen beleuchtete Fläche übrig. Vom Ergebnis der Schließung wird dann das Originalbild subtrahiert, und man erhält ein relativ homogen ausgeleuchtetes Bild; statt der Subtraktion kann auch eine Division verwendet werden. Natürlich ist es das Beste, die Beleuchtung bei der Aufnahme sorgfältig zu kontrollieren, jedoch gehen wir hier davon aus, dass das bestmögliche Bild zur Vorverarbeitung kommt.

2.5.6 Strichstärke

Die Bedeutung von Linienmustern ist in weiten Grenzen unabhängig von der *Strichstärke*, sodass es naheliegt, diese zunächst auf einen einheitlichen Wert, i. Allg. einen Rasterpunkt, zu normieren. Auch bei Schriftzeichen werden solche Verfahren immer wieder angewendet. Allerdings kann sich so eine Normierung je nach Klassifikationsverfahren auch nachteilig auswirken; dieses ist ein experimentell untermauertes Beispiel für das schon am Anfang von Kapitel 2 aufgezeigte Problem, dass der Erfolg einer Vorverarbeitungsoperation in der Regel im Zusammenhang mit den nachfolgenden Operationen beurteilt werden muss. Eine Linienverdünnung ist auch für die Kettencodierung zweckmäßig sowie für die Klassifikation von Fingerabdrücken und sonstigen Linienmustern.

Das Prinzip der Verfahren beruht darauf, in mehreren Durchgängen Randpunkte einer Linie abzuschälen, bis eine Linie, die nur einen Rasterpunkt dick ist, übrigbleibt. In den meisten Fällen werden einige oder alle der folgenden Forderungen gestellt:

1. Linien werden *nicht unterbrochen* und *nicht verkürzt*.
2. Die verdünnte Linie sollte etwa *in der Mitte* der ursprünglichen Linie liegen, auch wenn Bildstörungen vorliegen.
3. Das Verfahren muss *schnell* arbeiten.

Wichtigstes Element in der Linienverdünnung ist die Definition von Bedingungen für die Entfernung eines Bildpunktes. Ein Beispiel für solche Bedingungen beruht auf den in Bild 2.5.9 gezeigten *Masken* $M_1, \ldots, M_{19}$. Die Matrix $[f_{jk}]$ der Bildpunkte wird in vier disjunkte Teilmengen zerlegt, die bei der Linienverdünnung nacheinander spaltenweise bearbeitet werden. Zunächst werden nur die Masken $M_1, \ldots, M_{11}$ verwendet und ein Punkt P mit dem Wert 1 entfernt, wenn seine Nachbarn die durch die Masken festgelegten Werte haben. Dann wird mit

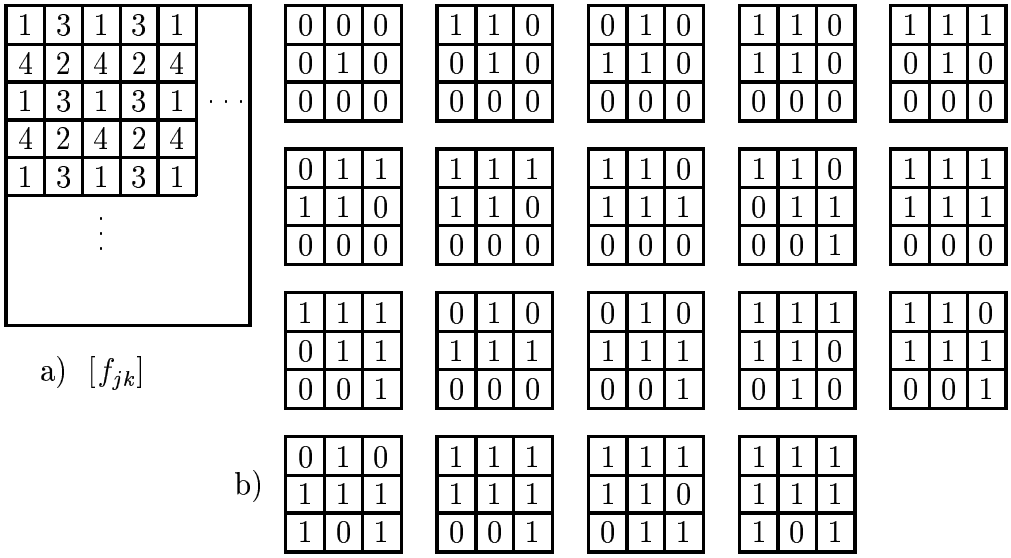


Bild 2.5.9: In a) ist die Zerlegung der Bildpunkte von $[f_{jk}]$ in vier disjunkte Teilmengen gezeigt, in b) sind die verwendeten Masken M_1 bis M_{19} von links nach rechts geordnet

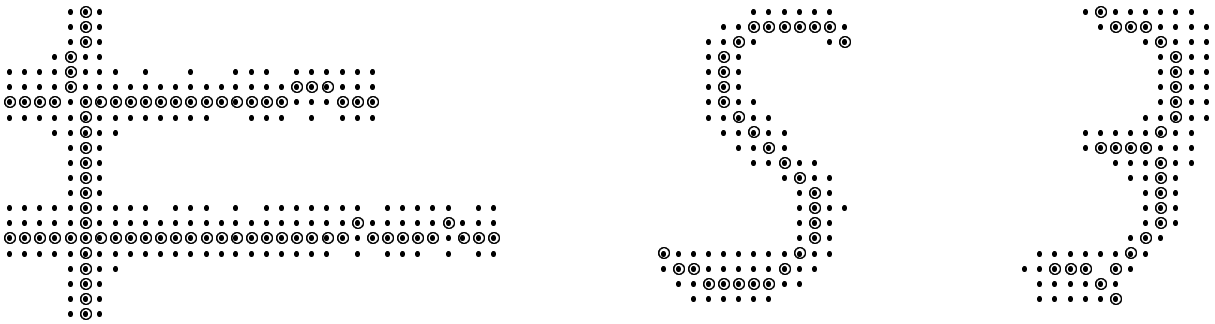


Bild 2.5.10: Ein Beispiel für die Verdünnung von Linien mit dem in Bild 2.5.9 skizzierten Verfahren

allen Masken weitergearbeitet. Stets werden auch die Konfigurationen geprüft, die aus den angegebenen Masken durch Spiegelung an der x– oder y–Achse oder durch Rotation um $90°, 180°$ und $270°$ Grad entstehen. In Bild 2.5.10 ist ein Beispiel für die Wirkung dieser Operationen angegeben.

Ein anderes Verfahren beruht ebenfalls auf der Untersuchung von 3×3 Nachbarschaften. Wie in (2.4.3) werden die acht Nachbarn des Punktes P mit $f_j^{(P)}, j = 0, 1, \ldots, 7$ bezeichnet. Ein Punkt P wird entfernt, wenn alle der folgenden Bedingungen zutreffen:

1. $$\sum_{j=0}^{7} |f_{j+1}^{(P)} - f_j^{(P)}| = \alpha \ , \quad \text{mit } \alpha = 0, 2 \text{ oder } 4 \ , \quad f_8^{(P)} = f_0^{(P)} \ , \tag{2.5.50}$$

2. $$\sum_{j=0}^{7} f_j^{(P)} \neq 1 \ , \tag{2.5.51}$$

3. $$f_0^{(P)} \wedge f_2^{(P)} \wedge f_4^{(P)} = 0 \text{ und } f_0^{(P)} \wedge f_2^{(P)} \wedge f_6^{(P)} = 0 \ ,$$

4. Wenn $\alpha = 4$, dann muss zusätzlich entweder 4.1 oder 4.2 erfüllt sein:

4.1 $f_0^{(P)} \wedge f_6^{(P)} = 1$ und $f_1^{(P)} \vee f_5^{(P)} = 1$ und
$f_2^{(P)} = f_3^{(P)} = f_4^{(P)} = f_7^{(P)} = 0$,

4.2 $f_0^{(P)} \wedge f_2^{(P)} = 1$ und $f_3^{(P)} \vee f_7^{(P)} = 1$ und
$f_1^{(P)} = f_4^{(P)} = f_5^{(P)} = f_6^{(P)} = 0$.

Für gleichmäßiges Arbeiten folgt auf einen Durchgang mit den Bedingungen 1–4 ein Durchgang mit den Bedingungen $1, 2, 5, 6$.

5. $f_2^{(P)} \wedge f_4^{(P)} \wedge f_6^{(P)} = 0$ und $f_4^{(P)} \wedge f_6^{(P)} \wedge f_0^{(P)} = 0$, (2.5.52)

6. Wenn $\alpha = 4$, dann muss zusätzlich entweder 6.1 oder 6.2 erfüllt sein:

6.1 $f_4^{(P)} \wedge f_2^{(P)} = 1$ und $f_5^{(P)} \vee f_1^{(P)} = 1$ und
$f_0^{(P)} = f_3^{(P)} = f_6^{(P)} = f_7^{(P)} = 0$,

6.2 $f_6^{(P)} \wedge f_4^{(P)} = 1$ und $f_7^{(P)} \vee f_3^{(P)} = 1$ und
$f_4^{(P)} = f_5^{(P)} = f_6^{(P)} = f_1^{(P)} = 0$.

Die Verfahren haben Probleme an Einmündungen und Kreuzungen von Linien wie aus Bild 2.5.10 ersichtlich ist. Das liegt an der kleinen verwendeten Nachbarschaft und lässt sich mit größeren Nachbarschaften verbessern. Es ist zu beachten, dass bei dem obigen Algorithmus für die Objekte eine 8–Nachbarschaft, für den Hintergrund eine 4–Nachbarschaft angenommen wurde (s.dazu auch Abschnitt 2.6.1) und dass die Bedingungen 1–3 bzw. $1, 2, 5$ auch bei 4–Nachbarschaften für Objekte gelten. Alle Algorithmen, die mit 3×3 Nachbarschaften arbeiten, lassen sich auf die Abfrage von Masken zurückführen. Es gibt nämlich für die acht Nachbarn eines Punktes P genau $2^8 = 256$ verschiedene Konfigurationen, die sich in geeigneten Masken definieren lassen. Unterschiede liegen lediglich in der Art der verwendeten Masken und der Reihenfolge ihrer Anwendung. Der erste Algorithmus lässt sich effektiv realisieren, wenn man die Nachbarn $(f_7^{(P)}, f_6^{(P)}, \ldots, f_0^{(P)})$ als Ziffern einer Binärzahl auffasst, deren Wert die Bedingung für das Löschen des Punktes P über eine Tabelle definiert.

2.5.7 Sprache

Durch individuelle Unterschiede im menschlichen Stimmtrakt werden Änderungen in der Aussprache von Worten und Lauten verursacht, die keinen Einfluss auf die Bedeutung haben. Eine Möglichkeit ist, solche sprecherbedingten Schwankungen möglichst vorher zu eliminieren. Die dabei angewendeten Verfahren beruhen auf einer Normierung von Eigenschaften des Frequenzspektrums der Laute. Eine andere und heute übliche Möglichkeit ist die Verwendung eines geeigneten Merkmalssatzes (s. Abschnitt 3.6) sowie das Training eines leistungsfähigen Klassifikators mit einer großen Stichprobe gesprochener Sprache von vielen Sprechern. Weiterhin führen Umgebungsgeräusche und die Übertragung von Sprache durch Telefonkanäle zu Störungen, die vor der Erkennung reduziert werden sollten. Die Verarbeitung der Abtastwerte f_j der Sprache erfolgt, wie auch in Abschnitt 3.6 erläutert, in Datenfenstern von z. B. 10ms Länge.

Standardparameter zur Klassifikation von Vokalen sind die ersten zwei bis vier Formanten, die man z. B. aus den Maxima des Modellspektrums eines Lautes bestimmen kann, wie auch in Abschnitt 3.5.4 kurz erläutert wird. Zum Beispiel werden für jeden der ersten beiden Formanten

die größte und die kleinste Frequenz aus einer Menge von Äußerungen verschiedener Vokale bestimmt. Alle Formantfrequenzen werden linear abgebildet, sodass die größte und kleinste Frequenz auf Normwerte fallen. Um die Eigenschaften eines Sprechers zu normieren, reichen bereits zwei bis drei Vokale, deren Formantfrequenzen die größten bzw. kleinsten Werte annehmen. Durch diese einfache Normierung wird eine wesentliche Reduktion der Streuung der Formantfrequenzen erreicht. Weitere Normierungsverfahren beruhen auf der Normierung der Formantfrequenzen mit Hilfe der Länge des Stimmtrakts und auf der Normierung des Spektrums mit einem zweipoligen inversen Filter.

Für die Reduktion von Störgeräuschen eignet sich die **spektrale Subtraktion**. Die Basis ist (2.3.38) für die additive Überlagerung von Signal und Störung, jedoch ohne Verzerrung durch ein lineares System. Aus der Gleichung

$$f_j = s_j + n_j \tag{2.5.53}$$

mit f_j dem gestörten beobachteten Muster, s_j dem idealen Signal und n_j der unbekannten additiven Störung (jeweils im aktuellen Block von Daten) folgt die Schätzgleichung im Frequenzbereich für die FOURIER-Transformierte des störungsreduzierten Signals zu

$$|\widehat{S}_{m,j}|^2 = |F_{m,j}|^2 - \alpha|\widehat{N}_{m,j}|^2 \; . \tag{2.5.54}$$

Die Subtraktion kann, wie hier angegeben, beim Betragsquadrat der FOURIER-Koeffizienten im m–ten Datenblock – dem Leistungsspektrum – beim Betrag der FOURIER-Koeffizienten oder bei den in Abschnitt 3.6 eingeführten mel–Koeffizienten durchgeführt werden. Eine Modifikation ist die Vermeidung negativer Werte und zu großer Änderungen durch die Gleichung

$$|\widehat{S}_{m,j}|^2 = \begin{cases} |F_{m,j}|^2 - \alpha|\widehat{N}_{m,j}|^2 & : \; |F_{m,j}|^2 - \alpha|\widehat{N}_{m,j}|^2 > \beta|F_{m,j}|^2 \\ \beta|F_{m,j}|^2 & : \; \text{sonst} \end{cases} \; . \tag{2.5.55}$$

Die Werte $|\widehat{S}_{m,j}|^2$ usw. sind die j–ten Koeffizienten im m–ten Block oder Fenster von Daten. Der Parameter β hat einen Wert von etwa 0,15. Schließlich wird als geglätteter Wert für $|\widehat{S}_{m,j}|^2$ der Median aus den Werten im vorigen, im aktuellen (m–ten) und im nachfolgenden Datenfenster verwendet. Ein Schätzwert der Störleistung wird gleitend in Sprachpausen berechnet nach

$$|\widehat{N}_{m,j}|^2 = (1-\gamma)|\widehat{N}_{m-1,j}|^2 + \gamma|F_{m,j}|^2 \; . \tag{2.5.56}$$

Dabei ist $\gamma \approx 0$ in Datenfenstern mit Sprache, $\gamma \approx 0,2$ in Sprachpausen. Dadurch werden zeitliche Änderungen in den Störungen mitgeführt.

Ein wirksames Verfahren zur Reduktion des Einflusses des Aufnahmenkanals ist der *cepstrale Mittelwertabzug*. Da dieser jedoch nicht auf dem Muster $\boldsymbol{f}$ sondern den Merkmalen $\boldsymbol{c}$ durchgeführt wird, wird er in Abschnitt 3.6.5 vorgestellt.

2.6 Operationen auf diskreten Mustern (VA.1.1.2, 30.12.2003)

In diesem Abschnitt wird kurz auf zwei allgemeine Ergebnisse über die Verarbeitung diskreter Muster eingegangen. Es handelt sich um die Definition eines zusammenhängenden Gebietes und um sequentielle und parallele Operationen.

2.6.1 Zusammenhang in diskreten Mustern

In Bild 2.6.1a ist ein Objekt auf einem Hintergrund gezeigt. Man sieht, dass das Objekt S, aber nicht der Hintergrund $\bar{S} = \bar{S}_1 \cup \bar{S}_2$ zusammenhängend ist. Wenn das Objekt abgetastet wird, ergibt sich bei einer bestimmten Schrittweite z. B. das in Bild 2.6.1b gezeigte Ergebnis. Um festzustellen, ob das diskrete Objekt zusammenhängend ist, muss dieser Begriff zunächst für den diskreten Fall definiert werden. Es seien f_{jk} und $f_{\mu\nu}$ zwei Bildpunkte, für die die beiden Abstände

$$\begin{aligned} d_1(f_{jk}, f_{\mu\nu}) &= |j-\mu| + |k-\nu| \ , \\ d_\infty(f_{jk}, f_{\mu\nu}) &= \max\{|j-\mu|, |k-\nu|\} \end{aligned} \tag{2.6.1}$$

definiert werden. Daraus folgt die Definition von geeigneten Nachbarschaften sowie des Zusammenhangs von Bildpunkten.

Definition 2.13 *Eine* 4**–Nachbarschaft** $N_4(f_{jk})$ *des Punktes* f_{jk} *ist die Menge der Bildpunkte* $f_{\mu\nu}$ *mit der Eigenschaft*

$$N_4(f_{jk}) = \{f_{\mu\nu} \mid d_1(f_{jk}, f_{\mu\nu}) \leq 1\} \ . \tag{2.6.2}$$

Entsprechend ist die 8**–Nachbarschaft** *definiert mit*

$$N_8(f_{jk}) = \{f_{\mu\nu} \mid d_\infty(f_{jk}, f_{\mu\nu}) \leq 1\} \ . \tag{2.6.3}$$

Definition 2.14 *Mit* S *wird eine beliebige Untermenge* $S \subseteq [f_{jk}]$ *der Bildmatrix bezeichnet. Die Menge* S *der Bildpunkte ist* **zusammenhängend**, *wenn es für jedes beliebige Paar* $f_{jk} \in S,\ f_{lm} \in S$ *eine Folge* $P_0, P_1, \ldots, P_n$ *von Bildpunkten gibt, sodass gilt*

$$\begin{aligned} &1. \quad P_i \in S, \quad i = 0, 1, \ldots, n \ , \\ &2. \quad P_0 = f_{jk} \text{ und } P_n = f_{lm} \ , \\ &3. \quad P_{i-1} \text{ und } P_i, \quad i = 1, \ldots, n \text{ sind benachbart} \ . \end{aligned} \tag{2.6.4}$$

Man kann zwei Punkte als benachbart bezeichnen, wenn sie entweder 4*–Nachbarn gemäß (2.6.2) oder* 8*–Nachbarn gemäß (2.6.3) sind. Im ersten Falle wird* S *als* 4–zusammenhängend, *im zweiten als* 8–zusammenhängend *bezeichnet.*

Legt man in Bild 2.6.1b eine 8–Nachbarschaft zugrunde, so ist das quantisierte Objekt zusammenhängend, genauer 8–zusammenhängend. Das ist auch intuitiv befriedigend, da das kontinuierliche Objekt ebenfalls zusammenhängend ist. Allerdings ist in diesem Sinne auch der Hintergrund zusammenhängend, und das widerspricht ganz krass der Anschauung. Im nichtquantisierten Bild ist nämlich der Hintergrund eindeutig nichtzusammenhängend, und auch im quantisierten Bild ist es ein Widerspruch, dass das von der geschlossenen Kurve S umgebene

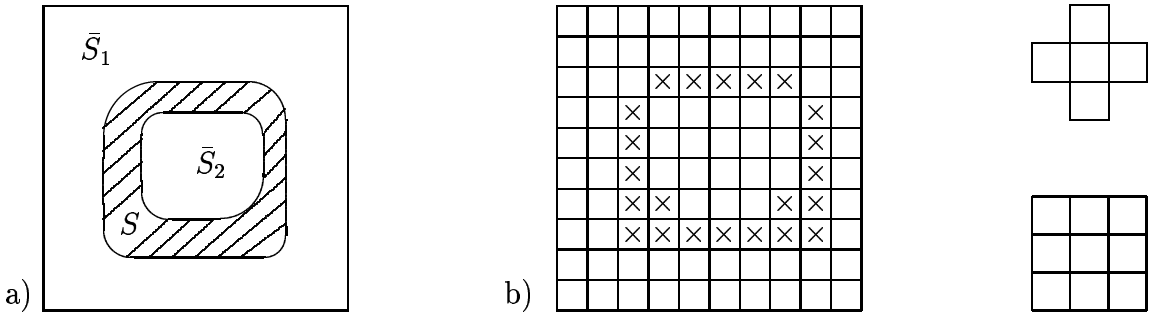


Bild 2.6.1: Ein Objekt S vor einem Hintergrund $\bar{S} = \bar{S}_1 \cup \bar{S}_2$, a) kontinuierlich und b) diskret; dazu rechts die 4– und die 8–Nachbarschaft

Gebiet $\bar{S}_1$ mit $\bar{S}_2$ zusammenhängen soll. Legt man eine 4–Nachbarschaft zugrunde, so wird zwar der Hintergrund nichtzusammenhängend, aber auch das Objekt S hängt nicht mehr zusammen. Man sieht an diesem Beispiel, dass die Übertragung von Begriffen aus dem kontinuierlichen in den diskreten Bereich mit Vorsicht zu geschehen hat. Der obige Widerspruch lässt sich beseitigen, wenn man für das Objekt oder die Punktmenge S eine 8–Nachbarschaft verwendet und für den Hintergrund oder das Komplement eine 4–Nachbarschaft. Ebenso ist es möglich, für S eine 4–Nachbarschaft und für $\bar{S}$ eine 8–Nachbarschaft zu verwenden.

2.6.2 Parallele und sequentielle Operationen

Operationen auf Folgen $[f_j]$ oder $[f_{jk}]$ von Abtastwerten können parallel oder sequentiell ausgeführt werden. Bei einer **parallelen Operation** werden für jeden Punkt als Operanden nur die ursprünglich gegebenen Werte von $[f_{jk}]$ verwendet. Ist T_P eine lokale parallele Operation, die z. B. als Operanden nur eine 8–Nachbarschaft jedes Punktes f_{jk} verwendet, so gilt

$$\begin{aligned} h_{jk} \;=\; & T_P\,\{f_{j-1,k-1},\; f_{j-1,k},\; f_{j-1,k+1},\; f_{j,k-1},\; f_{jk}, \\ & f_{j,k+1},\; f_{j+1,k-1},\; f_{j+1,k},\; f_{j+1,k+1}\}\;. \end{aligned} \tag{2.6.5}$$

Man kann sich dieses so vorstellen, dass T_P gleichzeitig oder *parallel* auf *alle* Elemente von $[f_{jk}]$ angewendet wird. Da entsprechende Rechner nach wie vor selten sind, wird T_P jeweils nur auf *ein* Element f_{jk} angewendet und das Ergebnis h_{jk} in einem getrennten Speicherbereich aufgehoben, um auch in allen folgenden Schritten stets die Ausgangsfolge $[f_{jk}]$ zur Verfügung zu haben.

Bei einer **sequentiellen Operation** T_S werden die Elemente von $[f_{jk}]$ in einer definierten Reihenfolge *nacheinander* abgearbeitet, wobei in jedem Schritt die Ergebnisse der vorigen Schritte verwendet werden. Wird als Reihenfolge vereinbart, dass zunächst der Index j, dann der Index k anwächst, also $[f_{jk}]$ zeilenweise bearbeitet wird, so ist

$$\begin{aligned} h_{jk} \;=\; & T_S\,\{h_{j-1,k-1},\; h_{j-1,k},\; h_{j-1,k+1},\; h_{j,k-1},\; f_{jk}, \\ & f_{j,k+1},\; f_{j+1,k-1},\; f_{j+1,k},\; f_{j+1,k+1}\}\;. \end{aligned} \tag{2.6.6}$$

In diesem Falle kann also das Ergebnis h_{jk} in $[f_{jk}]$ selbst gespeichert werden und der getrennte Speicherbereich für $[h_{jk}]$ entfällt. Allerdings steht $[f_{jk}]$ am Ende nicht mehr zur Verfügung. Für parallele und sequentielle lokale Operationen gilt

Satz 2.14 *Jede Transformation einer Folge $[f_{jk}]$, die durch eine Reihe paralleler lokaler Operationen bewirkt wird, kann auch durch eine Reihe sequentieller lokaler Operationen bewirkt werden, und umgekehrt.*
Beweis*: s. z. B. [Rosenfeld und Pfaltz, 1966].*

Der zum obigen Satz zitierte Beweis ist konstruktiv; er wird wegen seiner Länge hier nicht wiederholt. Es gilt allgemein:

1. Jede *parallele* lokale Operation ist äquivalent zu *zwei* sequentiellen lokalen Operationen.
2. Eine *sequentielle* lokale Operation kann *viele* parallele lokale Operationen erfordern.

Im Verlauf dieses Kapitels wurden Operationen ausschließlich in der parallelen Version angegeben. Wie erwähnt, heißt das nicht, dass man zu ihrer Realisierung einen Parallelrechner haben muss, er wäre jedoch vorteilhaft.

2.7 Literaturhinweise

Kodierung

Umfassende Darstellungen zu dem Thema findet man auch in den Büchern [Oppenheim und Schafer, 1975, Pratt, 1991]. Die Abtastung über ein endliches Intervall wird in [Pratt, 1991] behandelt, die Verwendung schiefwinkliger Koordinaten in [Rosenfeld und Kak, 1976], die Interpolation aus nicht äquidistanten Abtastwerten in [Arigovindan et al., 2005]. Die Beschleunigung der Suche nach einem Kodewort durch Arbeiten im Transformationsbereich (speziell der HAAR-Transformation) ist in [Hwang et al., 2000] beschrieben. Für Beweise des Abtasttheorems wird auch auf [Middleton, 1960, Winkler, 1977] verwiesen, die Beziehung für die Zahl der Amplitudenstufen findet man in [Jayant, 1974], die für die Quantisierungskennlinie in [Max, 1960, Rosenfeld und Kak, 1976]; dort werden auch nichtlineare Quantisierungskennlinien diskutiert. Die Charakterisierung der Bildqualität wird in [**?**] untersucht. Der LBG–Algorithmus zur Vektorquantisierung geht auf [Linde et al., 1980] zurück, weiteres Material dazu enthält [Gersho und Gray, 1992, Makhoul et al., 1985, Schukat-Talamazzini et al., 1993, Patane und Russo, 2001, Patane und Russo, 2002]. Zur Lauflängen–Kodierung wird auf [Gonzales und Wintz, 1977, Van Voorhis, 1976] verwiesen, für den Kettenkode auf [Freeman, 1961, Freeman, 1974, Morrin, 1976]. Zur Generierung von Linienzeichnungen aus Bildern wurden eine Reihe von Verfahren entwickelt [Deussen und Strothotte, 2000, Lansdown und Schofield, 1995, Salisbury et al., 1994, Winkenbach und Salesin, 1994]. Weiteres Material zur Kodierung, darunter optimale und fehlerkorrigierende Kodes, findet man z. B. in [Clarke, 1990, Jacquin, 1993, Jayant, 1976, Jayant und Noll, 1984, McWilliams und Sloane, 1978, Niemann und Wu, 1993].

Schwellwertoperationen

Übersichten über Schwellwertoperationen geben [Glasbey, 1993, Weszka, 1978, Sahoo et al., 1988]. Schwellwertoperationen im Frequenzbereich wurden in [Lee, 2001] eingeführt (jedoch nicht die Bestimmung des Schwellwertes). Zur Berechnung optimierter Schwellwerte wird auf [Otsu, 1978, Otsu, 1979] verwiesen und für weitere Hinweise auf [Murtag und Starck, 2003, Pun, 1980, Rosenfeld und Kak, 1976]. Die Klassenunsicherheit wird in [Brink und Pendock, 1996, Dunn et al., 1984, Kapur et al., 1985, Kittler und Illingworth, 1986, Leung und Lam, 1998] verwendet; das kombinierte Kriterium (2.2.27), S. 86 sowie die dabei verwendete Homogenitätsfunktion $h(f_{j,k})$ sind in [Saha und Udupa, 2001] zu finden (h wird dort als μ_φ bezeichnet). Eine Kombination lokaler und globaler Kriterien bietet auch [Cheng et al., 2002]. Auch Diffusionsfilter werden genutzt [Manay und Yezzi, 2003].

Beispiele für die Anwendung von Schwellwertoperationen zur Segmentierung von Angiogrammen, Schriftzeichen, Werkstücken, Zellen, Chromosomen, Blutgefäßen und Linien in Schaltplänen sind in [Bley, 1982, Chow und Kaneko, 1972b, Chow und Kaneko, 1972a, Gong et al., 1998, Jiang und Mojon, 2001, Schürmann, 1974, Agin, 1980, Ledley, 1964, Ingram und Preston, 1970, Kubitschek, 1979] enthalten. In [Ohlander et al., 1978] wird eine Folge von Schwellwertoperationen auch zur Segmentierung von Farbbildern benutzt.

Das gefilterte Histogramm wird in [Weszka et al., 1974] zur Schwellwertbestimmung eingeführt, Grauwerte auf der Konturlinie in [Wang und Bai, 2003]. Weitere Arbeiten dazu sind [Barrett, 1981, Bunke et al., 1982, Lim und Lee, 1990, Perez und Gonzalez, 1987] Das Problem unimodaler Histogramme behandeln [Bhanu und Faugeras, 1982, Rosin, 2001,

Tsai, 1995]. In [Rosin, 2001] finden sich Beispiele zu Ergebnissen mit Schwellwertoperationen für die Extraktion von Kantenelementen nach [Kapur et al., 1985, Ridler und Calvard, 1978, Otsu, 1979, Prager, 1980, Rosin, 2001, Tsai, 1985]. Histogramme sind auch die Basis einiger Verfahren zur Merkmalsgewinnung [Ferman et al., 2002].

Lineare Operationen

Lineare Systeme und Filterung werden für den kontinuierlichen Fall z. B. in [Winkler, 1977, Morrin, 1974, Guillemin, 1963] behandelt, für den digitalen Fall in [Oppenheim und Schafer, 1975, Oppenheim et al., 1983, Oppenheim und Schafer, 1989, Pratt, 1991, Schüßler, 1992]. In diesen Büchern wird auch auf die FOURIER-Transformation sowie auf die hier nicht behandelten Systeme mit unendlicher Ausdehnung der Impulsantwort eingegangen. Beziehungen zwischen kontinuierlicher und diskreter FOURIER-Transformation werden in [Bergland, 1969, Niemann, 1973, Niemann, 1981] diskutiert. Zur schnellen FOURIER-Transformation (FFT) wird weiter auf [Bergland, 1969, Brigham, 1995, Brigham, 1997, Cochran et al., 1967, Cooley und Tukey, 1965, Duhamel und Vetterli, 1990, Pease, 1968] verwiesen; [Frigo und Johnson, 1998] beschreibt eine effiziente Implementierung. Eine (hier nicht behandelte) Verallgemeinerung der FOURIER-Transformation durch Einführung eines Ordnungsparameters (“fractional Fouriertransform”) wird ausführlich in [Ozaktas et al., 2001] diskutiert, ein Sonderheft dazu ist [Ortigueira und Machado, 2003].

Die Synthese linearer Systeme bzw. Filter wird in [Lacroix, 1980, Schüßler, 1973, Schüßler, 1992, Temes und Mitra, 1973] behandelt, in [Jähne et al., 1999b] die Besonderheiten der Synthese von Filtern für die Bildverarbeitung. Außer der schnellen FOURIER-Transformation wurden weitere schnelle Algorithmen für die Signalverarbeitung entwickelt, z. B. [Agarwal und Burrus, 1974, Alshibami et al., 2001, Ramesh und Burrus, 1974, Reed und Truong, 1975, Rubanov et al., 1998]. Das rekursive GAUSS-Filter in (2.3.64), (2.3.65) ist im Einzelnen in [Young und van Vliet, 1995] beschrieben, das anisotrope GAUSS-Filter in [Geusebroek et al., 2003]. Weitere rekursive Filter sind z. B. [Deriche, 1990] zu entnehmen, einen Ansatz zur Approximation von FIR durch IIR–Filter gibt [Brandenstein und Unbehauen, 1998]. Die Nutzung des KALMAN-Filter wird in [Angwin und Kaufman, 1989, Biemanond und Gerbrands, 1979, Evensen, 2003, Woods und Radewan, 1977, Woods und Ingle, 1981] untersucht.

Besondere Aufmerksamkeit haben lineare Filter auch für die Ermittlung von Kanten in Bildern gefunden [Bourennane et al., 2002, Canny, 1986, Demigny, 2002, Deriche, 1987, Deriche, 1990, Deriche und Giraudon, 1993, Petrou und Kittler, 1991, Prewitt, 1970, Shen und Castan, 1986, Shen und Zhao, 1990, Shen, 1992].

Zum allgemeinen Problem der Restauration wird auf [Andrews und Hunt, 1975, Batex und McDonnel, 1986, Katsaggelos, 1991, Pratt, 1991] verwiesen. Überblicke über die Verbesserung von Bildfolgen geben [Brailean et al., 1995, Kaufman und Tekalp, 1991].

Ein Teilraumansatz zur Verbesserung von Sprache wird in [Hu und Loizou, 2003a] entwickelt, ein den Maskierungseffekt nutzender in [Hu und Loizou, 2003b].

Nichtlineare Operationen

Ein Beispiel für binäre Maskenoperationen gibt [Rao, 1976], die Glättung von Linienmustern [Ehrich, 1978, Hanaki et al., 1976, Mason und Clemens, 1968, Niemann, 1974], speziell im Kettenkode [Sklansky und Nahin, 1972, Eccles und Rosen, 1977].

Rangordnungsfilter werden in [Arce und Foster, 1989, Gallagher Jr. und Wise, 1981, Herp et al., 1980, Herp, 1980, Heygster, 1979, Hodgson et al., 1985] behandelt, Stack–Filter in [Coyle et al., 1989, Lin und Coyle, 1990, Wendt et al., 1986].

Eine Variante des Medianfilters, s. [Borda und Frost, 1968, Rabiner et al., 1975, Huang et al., 1979], ist der mehrstufige Median [Arce und Foster, 1989, Nieminen et al., 1987] und der zentralgewichtete Median [Ko und Lee, 1991, Yin et al., 1996]. Medianfilter für vektorwertige Muster, insbesondere auch Farbbilder, wurden in [Astola et al., 1990, Reggazoni und Teschioni, 1997, Tang et al., 1995, Vardavoulia et al., 2001, Zheng et al., 1993] vorgeschlagen. Das in [Şenel et al., 2002] eingeführte topologische Medianfilter hat gute Eigenschaften bezüglich Störungsreduktion und Kantenerhaltung. Die Kontrastverstärkung geht auf [Kramer und Bruckner, 1975] zurück, die Konturhervorhebung wird in [Herp et al., 1980, Herp, 1980] genutzt. Impulsartige Störungen werden in [Abreu et al., 1996, Garnett et al., 2005, Pok und Liu, 2003, Wang und Zhang, 1999] betrachtet.

Als Vorläufer morphologischer Operationen sind z. B. [Golay, 1969, Preston, 1971] zu sehen. Morphologische Operationen werden in [Haralick et al., 1987, Jungmann, 1984, Maragos und Shafer, 1987a, Maragos und Shafer, 1987b, Mukhopadhyay und Chanda, 2002, Serra, 1982, Serra, 1986, Serra, 1988, Serra und Soille, 1994, Soille, 1999a, Sternberg, 1986, Yang und Li, 1995] behandelt, ihr Entwurf in [Hyonam et al., 1990, Li et al., 1996, Vogt, 1989]; Übersichten geben [Haralick et al., 1987, Sternberg, 1986, Soille, 1999b]. Eine kurze Übersicht über den Bezug zu Rangordnungsfiltern ist in [Soille, 2002] gegeben. In [Soille, 1999b] werden auch die Zusammenhänge zu Abstandstransformationen diskutiert sowie die Verwendung von *zwei* Eingabebildern; solches findet man z. B. auch in [Levi und Montanari, 1970, Preston, 1971]. Effiziente Algorithmen für morphologische Operationen werden in [van den Boomgaard und van Balen, 1992, Breen und Monro, 1994, Breen und Jones, 1996, Gilboa et al., 2002, van Herk, 1992, Nacken, 1996, Paulus et al., 1995, Ragnemalm, 1992, Soille et al., 1996, Vincent, 1991, Vincent und Soille, 1991, Vincent, 1993] entwickelt.

In [Agaian et al., 2001] werden sequenzgeordnete Transformationen zur Verbesserung von Mustern (bzw. zum "image enhancement") eingeführt, die hier nicht behandelt wurden.

Diffusionsfilter wurden in [Perona und Malik, 1990] eingeführt und werden in [Weickert, 1998, Weickert et al., 1998, Weickert, 1999b] verallgemeinert und ausführlich insbesondere für Grauwertbilder, also den zweidimensionalen Fall, dargestellt; Verallgemeinerungen auf den dreidimensionalen Fall erfolgen in [Gering et al., 1992, Weickert, 1999a], Verallgemeinerungen auf vektorwertige Bilder in [Tschumperlé und Deriche, 2002]. Zum Ergebnis in (2.4.22), S. 115, wird auf [Hellwig, 1977, Petrovski, 1955] verwiesen. Schwellwertoperation und Segmentierung mit Diffusionsfiltern wird in [Manay und Yezzi, 2003] durchgeführt. Ein auf Waveletapproximation und Diffusion basierender Ansatz zur detailerhaltenden Glättung wurde in [Shih et al., 2003] entwickelt, ein weiterer waveletbasierter in [Figueiredo und Nowak, 2001]. Complexe Diffusionsfilter und (adaptive) Filter, auch zur Reduktion von "speckle"-Rauschen, wurden in [Chang et al., 2000, Geman und Geman, 1984, Frost et al., 1982, Gilboa et al., 2004, Kuan et al., 1987, Tekalp et al., 1989, Yu und Acton, 2002] entwickelt. Variationsansätze wurden hier ausgelassen; sie wurden in [Mumford und Shah, 1989] eingeführt und werden z. B. in [Geiger und Yuille, 1991, Heers et al., 2001, Morel und Solimini, 1995, Schnörr, 1998, Tsai et al., 2001] behandelt sowie im Sonderheft [Caselles et al., 1998]. Weitere Ansätze geben [Barcelos et al., 2003, Gilboa et al., 2002, Vermaak et al., 2002].

Der HARRIS-Operator zur Ermittlung interessanter Punkte in Bildern wurde in [Harris und Stephens, 1988] eingeführt und u. a. in [Heidemann, 2005, Schmid et al., 2000] zusammen mit anderen Operatoren untersucht. Weitere Operatoren wurden in [Dreschler, 1981, Förstner und Gülch, 1987, Frantz et al., 1998, Mikolajczyk und Schmid, 2004, Mikolajczyk und Schmid, 2005, Montesinos et al., 1998, Rohr, 1997, Smith und Brady, 1997, Schmid et al., 2000, Tian et al., 2001] vorgeschlagen und untersucht.

Weitere nichtlineare Ansätze, die hier nicht behandelt wurden, sind die in [Yaroslavski, 1997] entwickelten auf der diskreten cosinus-Transformation beruhenden Filter, die in [Chambolle et al., 1998] behandelten Wavelet basierten Verfahren ("wavelet shrinkage") sowie deren Kombination in hybriden Verfahren [Shaick et al., 2000] und die kernbasierten Ansätze [Kwok und Tsang, 2004, Mika et al., 1999]. Für das spezielle Problem der MOIRÉ-Störungen stehen ebenfalls Filter zur Verfügung [Yang und Tsai, 1998, Yang und Tsai, 2000].

Normierungsmaßnahmen

Auf die Interpolation mit B-Splines geht insbesondere [Catmull und Rom, 1974, Hou und Andrews, 1978, Unser et al., 1991, Unser et al., 1993a, Unser et al., 1993b] ein, auf den zu erwartenden Interpolationsfehler [Blu und Unser, 1999a, Blu und Unser, 1999b]. Die Probleme der Skalierung, Rotation und Interpolation von Mustern werden in [Harris, 1978, Hou und Andrews, 1978, Jähne, 2000, Lehmann et al., 1999, Meijering et al., 2001, Muñoz et al., 2001, Thévenaz et al., 2000a, Thévenaz et al., 2000b, Unser et al., 1995, Unser, 1999] behandelt. Ein allgemeiner Ansatz zur Generierung von Interpolationskernen wird in [Blu et al., 2003] vorgestellt, positiv definite Funktionen in [Wendland, 1995]. Frühere Arbeiten zur Interpolation sind z. B. [Greville, 1944, Karup, 1899, Henderson, 1906]. Weitere Ansätze zur Interpolation sind in [Hladůvka und Gröller, 2001, Kaup et al., 2005, Xu et al., 2000] beschrieben.

Die Nützlichkeit von Größennormierungen für die Klassifikation von Schriftzeichen und gesprochenen Wörtern wurde schon in [Güdesen, 1976, Martin, 1977, Schürmann, 1978] nachgewiesen. Die Lagenormierung mit Momenten wurde in [Paton, 1970] verwendet; etwas andere Operationen werden in [Alt, 1962, Nagel und Rosenfeld, 1977, Kwag et al., 2002] angegeben, die auch eine Scherung von Mustern bewirken. Auf weitere Arbeiten, insbesondere auch zur Berechnung von Momenten, wird in Abschnitt 3.11 verwiesen.

Übersichten über die (hier nicht behandelte) Registrierung von Bildern geben [Brown, 1992, Lester und Arridge, 1999, Maintz und Viergever, 1998]. Weitere Arbeiten zu unterschiedlichen Aspekten der Bildregistrierung sind [Chen et al., 1999, Gee, 1999, Musse et al., 1999] sowie andere Arbeiten in dem Sonderheft [Goshtasby und Le Moigne, 1999].

Die Energienormierung wird z. B. in [Regel, 1982, Schafer und Rabiner, 1975, Silverman und Dixon, 1974] verwendet, weitere Fensterfunktionen findet man in Sect. 5.5 von [Oppenheim und Schafer, 1975]. Morphologische Operationen zur Normierung findet man in [Soille, 1999b, Soille, 1999a].

Die Verdünnung von binären Linienobjekten wird für Schriftzeichen in [Agui und Nagahashi, 1979, Ahmed und Ward, 2002, Güdesen, 1976, Rieger, 1979, Rieger, 1979, Triendl, 1970], für Fingerabdrücke und sonstige Linienzeichnungen in [Kreifelts, 1977, Rao, 1976, Zhou et al., 1995], für 3D–Objekte in [Ma und Wan, 2001] und u. a. für Angiogramme in [Rockett, 2005] (als verbesserte Version von [Ahmed und Ward, 2002]) behandelt; Übersichten geben [Ablamayko und Pridmore, 2000,

Deutsch, 1972, Lam et al., 1992, Lam et al., 1993, Lam und Suen, 1995, Rosenfeld, 1975, Stefanelli und Rosenfeld, 1971]. Die in Bild 2.5.9 angegebenen Masken sind aus [Rosenfeld, 1975], die Bedingungen (2.5.50) und (2.5.52) aus [Deutsch, 1972]. Größere Nachbarschaften werden in [Rieger, 1979, Murthy und Udupa, 1974] verwendet.

Frühe Arbeiten zur Normierung von Formantfrequenzen sind [Gerstman, 1968, Itakura, 1975, Wakita, 1977, Wakita, 1979]. Arbeiten zur spektralen Subtraktion sind [Berouti et al., 1979, Boll, 1979, Gustafsson et al., 2001, Hirsch und Ehrlicher, 1995, Klemm, 1994, Kushner et al., 1989]. Eine umfassende Darstellung zu robusten Verfahren gibt [Junqua und Haton, 1996]. Umfangreiche experimentelle Untersuchungen zu verschiedenen Verfahren der Sprachnormierung wurden in [Jaschul, 1982, Kämmerer, 1989, Schless, 1999] durchgeführt, die Vokaltraktnormierung in [Eide und Gish, 1996, Wakita, 1977, Welling et al., 2002] untersucht. Eine Übersicht über Verfahren der Störungs– und Echoreduktion gibt [Le Bouquin Jeannès et al., 2001].

Weitere Normierungsmaßnahmen für Bilder, wie Histogrammlinearisierung, Farbnormierung oder geometrische Korrekturen sind in [Niemann, 1990] dargestellt. Vorschläge und Vergleiche zur Farbnormierung und –invarianten enthalten [Barnard et al., 2002a, Barnard et al., 2002b, Finlayson et al., 2001, Geusebroek et al., 2001, Lehmann und Palm, 2001].

Operationen auf diskreten Mustern

Die Ergebnisse zum Zusammenhang von Mustern gehen auf [Rosenfeld, 1970, Rosenfeld, 1979] zurück. Eine Darstellung der Konzepte digitaler Geometrie gibt [Klette, 2001], der digitalen Topologie [Kong, 2001].

Literaturverzeichnis

[Ablamayko und Pridmore, 2000] Ablamayko, S., Pridmore, T. *Machine Interpretation of Line Drawing Images*. Springer, Berlin Heidelberg, 2000.

[Abreu et al., 1996] Abreu, E., Lightstone, M., Mitra, S., Arakawa, K. A new efficient approach for the removal of impulse noise from highly corrupted images. *IEEE Trans. on Image Processing*, 5:1012–1025, 1996.

[Agaian et al., 2001] Agaian, S.S., Panetta, K., Grigoryan, A.M. Transform-based image enhancement algorithms with performance measure. *IEEE Trans. on Information Theory*, 47:367–382, 2001.

[Agarwal und Burrus, 1974] Agarwal, R.C., Burrus, C.S. Number theoretic transforms to implement fast digital convolution. *Proc. IEEE*, 63(4):550–560, 1974.

[Agin, 1980] Agin, G.J. Computer vision systems for industrial inspection and assembly. *Computer*, 13(5):11–20, 1980.

[Agui und Nagahashi, 1979] Agui, T., Nagahashi, H. A description method of handprinted Chinese characters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:20–24, 1979.

[Ahmed und Ward, 2002] Ahmed, M., Ward, R. A rotation invariant rule-based thinning algorithm for character recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1672–1678, 2002.

[Alshibami et al., 2001] Alshibami, O., Boussatka, S., Aziz, M. Fast algorithm for the 2-d new Mersenne number transform. *Signal Processing*, 81:1725–1735, 2001.

[Alt, 1962] Alt, F.L. Digital pattern recognition by moments. In G.L. Fischer, D.K. Pollock, B. Raddack, M.E. Stevens, Hg., *Optical Character Recognition*, S. 153–179. Spartan Books, Washington, 1962.

[Andrews und Hunt, 1975] Andrews, H.C., Hunt, B.R. *Digital Image Restoration*. Prentice Hall, Englewood Cliffs, N.J., USA, 1975.

[Angwin und Kaufman, 1989] Angwin, D., Kaufman, H. Image restoration using a reduced order model Kalman filter. *IEEE Trans. on Signal Processing*, 16:21–28, 1989.

[Arce und Foster, 1989] Arce, G.R., Foster, R.E. Detail preserving rank-order based filters for image processing. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37:83–98, 1989.

[Arigovindan et al., 2005] Arigovindan, M., Sühling, M., Hunziker, P., Unser, M. Variational image reconstruction from arbitrarily spaced samples: A fast multiresolution spline solution. *IEEE Trans. on Image Processing*, 14:450–460, 2005.

[Astola et al., 1990] Astola, J., Haavisto, P., Neuvo, Y. Vector median filters. *Proc. IEEE*, 78:678–689, 1990.

[Barcelos et al., 2003] Barcelos, C.A.Z., Boaventura, M., Silva, E.C. A well-balanced flow equation for noise removal and edge detection. *IEEE Trans. on Image Processing*, 14:751–763, 2003.

[Barnard et al., 2002a] Barnard, K., Cardei, V., Funt, B. A comparison of computational color constancy algorithms – Part I: Methodology and experiments with synthesized data. *IEEE Trans. on Image Processing*, 11:972–984, 2002a.

[Barnard et al., 2002b] Barnard, K., Martin, L., Coath, A., Funt, B. A comparison of computational color constancy algorithms – Part II: Experiments with image data. *IEEE Trans. on Image Processing*, 11:985–996, 2002b.

[Barrett, 1981] Barrett, W.A. An iterative algorithm for multiple threshold detection. In *Conf. on Pattern Recognition and Image Processing*, S. 273–278. IEEE Comp. Soc., Dallas TX, 1981.

[Batex und McDonnel, 1986] Batex, R.H.T., McDonnel. *Image Restoration and Reconstruction*. Clarendon Press, Oxford, 1986.

[Bergland, 1969] Bergland, G.D. A guided tour of the fast Fourier transform. *IEEE Spectrum*, 6(7):41–52, 1969.

[Berouti et al., 1979] Berouti, M., Schwartz, R., Makhoul, J. Enhancement of speech corrupted by acoustic noise. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 208–211. Washington DC, 1979.

[Bhanu und Faugeras, 1982] Bhanu, B., Faugeras, O.D. Segmentation of images having unimodal distributions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 4:408–419, 1982.

[Biemanond und Gerbrands, 1979] Biemanond, J., Gerbrands, J. An edge preserving recursive noise smoothing algorithm for image data. *IEEE Trans. on Systems, Man, and Cybernetics*, 9:622–627, 1979.

[Bley, 1982] Bley, H. *Vorverarbeitung und Segmentierung von Stromlaufplänen unter Verwendung von Bildgraphen*. Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 1982.

[Blu et al., 2003] Blu, T., Thevenaz, P., Unser, M. Complete parameterization of piecewise-polynomial kernels. *IEEE Trans. on Image Processing*, 12:1297–1309, 2003.

[Blu und Unser, 1999a] Blu, T., Unser, M. Quantitative Fourier analysis of approximation techniques. Part I – interpolators and projectors. *IEEE Trans. on Signal Processing*, 47:2783–2795, 1999a.

[Blu und Unser, 1999b] Blu, T., Unser, M. Quantitative Fourier analysis of approximation techniques. Part II — wavelets. *IEEE Trans. on Signal Processing*, 47:2796–2806, 1999b.

[Boll, 1979] Boll, S. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 27:113–120, 1979.

[Borda und Frost, 1968] Borda, R.P., Frost, J.D. Error reduction in small sample averaging through the use of the median rather than the mean. *Electroenceph. clin. Neurophysiol.*, 25:391–392, 1968.

[Bourennane et al., 2002] Bourennane, E., Gouton, P., Paindavoine, M., Truchetet, F. Generalization of Canny-Deriche filter for detection of noisy exponential edge. *Signal Processing*, 82:1317–1328, 2002.

[Brailean et al., 1995] Brailean, J.C., Kleihorst, R.C., Efstratiadis, S.N., Katsaggelos, A.K., Langedijk, R.L. Noise reduction filters for dynamic image sequences: A review. *Proc. IEEE*, 83:1272–1292, 1995.

[Brandenstein und Unbehauen, 1998] Brandenstein, H., Unbehauen, R. Least-squares approximation of FIR by IIR digital filters. *IEEE Trans. on Signal Processing*, 46:21–30, 1998.

[Breen und Jones, 1996] Breen, E., Jones, R. Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, 1996.

[Breen und Monro, 1994] Breen, E., Monro, D. An evalutation of priority queues for mathematical morphology. In [Serra und Soille, 1994], S. 249–256.

[Brigham, 1995] Brigham, E.O. *FFT Schnelle Fourier-Transformation*. R. Oldenbourg Verlag, München, Germany, 6. Aufl., 1995.

[Brigham, 1997] Brigham, E.O. *FFT Anwendungen*. R. Oldenbourg Verlag, München, Germany, 6. Aufl., 1997.

[Brink und Pendock, 1996] Brink, A.D., Pendock, N.E. Minimum cross-entropy threshold selection. *Pattern Recognition*, 29:179–188, 1996.

[Brown, 1992] Brown, L.G. A survey of image registration techniques. *ACM Computing Surveys*, 24:325–376, 1992.

[Bunke et al., 1982] Bunke, H., Feistel, H., Niemann, H., Sagerer, G., Wolf, F., Zhou, G. X. Smoothing, thresholding, and contour extraction in images from gated blood pool studies. In *Proc. First IEEE Computer Society Int. Symp. in Medical Imaging and Image Interpretation*, S. 146–151. Berlin,

Germany, 1982.

[Canny, 1986] Canny, J. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[Caselles et al., 1998] Caselles, V., Morell, J.-M., Sapiro, G., Tannenbaum, A. Introduction to the special issue on partial differential equations and geometry-driven diffusion in image processing. *IEEE Trans. on Image Processing*, 7(3):269–273, 1998.

[Catmull und Rom, 1974] Catmull, E., Rom, R. A class of local interpolating splines. In R.E. Barnhill, R.F. Riesenfeld, Hg., *Computer Aided Geometric Design*, S. 317–326. Academic Press, New York, 1974.

[Catte et al., 1992] Catte, F., Lions, P.-L., Morel, J.-M., Coll, T. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29:182–193, 1992.

[Chambolle et al., 1998] Chambolle, A., DeVore, R.A., Lee, N., Lucier, B.J. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. on Image Processing*, 7:319–335, 1998.

[Chang et al., 2000] Chang, S.G., Yu, B., Vetterli, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. on Image Processing*, 9:1532–1546, 2000.

[Chen et al., 1999] Chen, M., Kanade, T., Pomerleau, D., Rowley, H.A. Anomaly detection through registration. *Pattern Recognition*, 32:113–128, 1999.

[Cheng et al., 2002] Cheng, H.D., Jiang, X.H., Wang, J. Color image segmentation based on homogram thresholding and region merging. *Pattern Recognition*, 35:373–393, 2002.

[Chow und Kaneko, 1972a] Chow, C.K., Kaneko, T. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research*, 5:388–410, 1972a.

[Chow und Kaneko, 1972b] Chow, C.K., Kaneko, T. Boundary detection of radiographic images by a threshold method. In S. Watanabe, Hg., *Frontiers of Pattern Recognition*, S. 61–82. Academic Press, New York, 1972b.

[Clarke, 1990] Clarke, R.J. *Transform Coding of Images*. Academic Press, New York, 1990.

[Cochran et al., 1967] Cochran, W.T., Cooley, J.W., Favin, D.L., Helms, H.D., Kaenel, R.A., Lang, W.W., Maling, G.C., Nelson, D.E., Rader, C.M., Welch, P.D. What is the fast Fourier transform? *Proc. IEEE*, 55:1664–1674, 1967.

[Cooley und Tukey, 1965] Cooley, J.W., Tukey, J.W. An algorithm for the machine computation of the complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[Courant und Hilbert, 1953] Courant, R., Hilbert, D. *Methods of Mathematical Physics, Vol. I*. Interscience Publishers, New York, 1953.

[Coyle et al., 1989] Coyle, E.J., Lin, J.-H., Gabbouj, M. Optimal stack filtering and the estimation and structural approaches to image processing. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37:2037–2066, 1989.

[Demigny, 2002] Demigny, D. On optimal linear filtering for edge detection. *IEEE Trans. on Image Processing*, 11:728–737, 2002.

[Deriche, 1987] Deriche, R. Using Canny's criteria to derive a recursively implemented optimal edge detector. *Int. Journal of Computer Vision*, 2:167–187, 1987.

[Deriche, 1990] Deriche, R. Fast algorithms for low-level vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:78–87, 1990.

[Deriche und Giraudon, 1993] Deriche, R., Giraudon, G. A computational approach for corner and vertex detection. *Int. Journal of Computer Vision*, 10:101–124, 1993.

[Deussen und Strothotte, 2000] Deussen, O., Strothotte, T. Computer generated pen-and-ink illustration of trees. In *Proc. SIGGRAPH*, S. 13–18. New Orleans, USA, 2000.

[Deutsch, 1972] Deutsch, E.S. Thinning algorithms on rectangular, hexagonal, and triangular arrays. *Communic. of the Association for Computing Machinery*, 15:827–837, 1972.

[Dreschler, 1981] Dreschler, L. *Ermittlung markanter Punkte auf den Bildern bewegter Objekte und Berechnung einer 3D-Beschreibung auf dieser Grundlage*. Dissertation, Fachbereich Informatik,

Universität Hamburg, Hamburg, Germany, 1981.

[Duhamel und Vetterli, 1990] Duhamel, P., Vetterli, M. Fast Fourier transforms: A tutorial review and a state of the art. *Signal Processing*, 19:259–299, 1990.

[Dunn et al., 1984] Dunn, S.M., Harwood, D., Davis, L.S. Local estimation of the uniform error threshold. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:742–747, 1984.

[Eccles und Rosen, 1977] Eccles, M.J.and McQueen, M.P.C., Rosen, D. Analysis of the digitized boundaries of planar objects. *Pattern Recognition*, 9:31–41, 1977.

[Ehrich, 1978] Ehrich, R.W. A symmetrical hysteresis smoothing algorithm that preserves principal features. *Computer Graphics and Image Processing*, 8:121–126, 1978.

[Eide und Gish, 1996] Eide, E., Gish, H. A parametric approach to vocal tract length normalization. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 346–349. Atlanta, USA, 1996.

[Evensen, 2003] Evensen, G. The ensemble Kalman filter: Theoreticel formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[Ferman et al., 2002] Ferman, A.M., Tekalp, A.M., Mehrotra, R. Robust color histogram descriptors for video segment retrieval and identification. *IEEE Trans. on Image Processing*, 11:497–508, 2002.

[Figueiredo und Nowak, 2001] Figueiredo, M.A.T., Nowak, R.D. Wavelet-based image estimation: An empirical Bayes approach using Jeffrey's noninformative prior. *IEEE Trans. on Image Processing*, 10:1322–1331, 2001.

[Finlayson et al., 2001] Finlayson, G.D., Hordley, S.D., Hubel, P.M. Color by correlation: A simple, unifying framework for color constancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:1209–1221, 2001.

[Förstner und Gülch, 1987] Förstner, W., Gülch, E. A fast operator for detection and precise location of distinct points, corners, and circular features. In *Proc. Intercommission Conference on Fast Processing of Photogrammetric Data*, S. 281–305. Interlaken, 1987.

[Frantz et al., 1998] Frantz, S., Rohr, K., Stiehl, H.S. Multi-step differential approaches for the localization of 3d point landmarks in medical images. *Journal of Computing and Information Technology*, 6:435–447, 1998.

[Freeman, 1961] Freeman, H. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Computers*, 10:260–268, 1961.

[Freeman, 1974] Freeman, H. Computer processing of line drawing images. *ACM Computing Surveys*, 6:57–97, 1974.

[Frigo und Johnson, 1998] Frigo, M., Johnson, S.G. FFTW: An adaptive software architecture for the FFT. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 3, 1381–1384. Seattle, Washington, 1998.

[Frost et al., 1982] Frost, V.S., Stiles, J.A., Shanmugan, K.S., Holtzman, J.C. A model of radar images and its application to adaptive digital filtering of multiplicative noise. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 4:157–165, 1982.

[Gallagher Jr. und Wise, 1981] Gallagher Jr., N.C., Wise, G.L. A theoretical analysis of the properties of median filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 29:1136–1141, 1981.

[Garnett et al., 2005] Garnett, R., Huegerich, H., Chui, C., He, W. A universal noise removal algorithm with an impulse detector. *IEEE Trans. on Image Processing*, 14:1747–1754, 2005.

[Gee, 1999] Gee, J.C. On matching brain volumes. *Pattern Recognition*, 32:99–111, 1999.

[Geiger und Yuille, 1991] Geiger, D., Yuille, A. A common framework for image segmentation. *Int. Journal of Computer Vision*, 6(3):227–243, 1991.

[Geman und Geman, 1984] Geman, S., Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[Gering et al., 1992] Gering, G., Kübler, O., Kikinis, R., Jolesz, F. Nonlinear isotropic filtering of MR data. *IEEE Trans. on Medical Imaging*, 11:221–232, 1992.

[Gersho und Gray, 1992] Gersho, A., Gray, R.M. *Vector Quantization and Signal Compression*. Kluwer, Boston, 1992.

[Gerstman, 1968] Gerstman, L.J. Classification of self-normalized vowels. *IEEE Trans. on Audio Electroacoustics*, 16:78–80, 1968.

[Geusebroek et al., 2003] Geusebroek, J.-M., Smeulders, A.W.M., van de Weijer, J. Fast anisotropic Gauss filtering. *IEEE Trans. on Image Processing*, 12:938–943, 2003.

[Geusebroek et al., 2001] Geusebroek, J.-M., van den Boomgaard, R., Smeulders, A.W.M., Geerts, H. Color invariance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:1338–1350, 2001.

[Gilboa et al., 2002] Gilboa, G., Sochen, N., Zeevi, Y.Y. Forward-backward diffusion processes for adaptive image enhancement and denoising. *IEEE Trans. on Image Processing*, 11:689–703, 2002.

[Gilboa et al., 2004] Gilboa, G., Sochen, N., Zeevi, Y.Y. Image enhancement and denoising by complex diffusion processes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1020–1036, 2004.

[Glasbey, 1993] Glasbey, C.A. An analysis of histogram-based thresholding algorithms. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 55(6), 1993.

[Golay, 1969] Golay, M.J.E. Hexagonal parallel pattern transformations. *IEEE Trans. on Computers*, 18:733–740, 1969.

[Gong et al., 1998] Gong, J.A., Li, L.Y., Chen, W.N. Fast recursive algorithms for 2-dimensional thresholding. *Pattern Recognition*, 31:295–300, 1998.

[Gonzales und Wintz, 1977] Gonzales, R.C., Wintz, P. *Digital Image Processing*. Addison-Wesley, Reading, Mass., 1977.

[Goshtasby und Le Moigne, 1999] Goshtasby, A.A., Le Moigne, J. Image registration: Guest Editor's introduction. *Pattern Recognition*, 32(1):1–2, 1999.

[Greville, 1944] Greville, T.N.E. The general theory of osculatory interpolation. *Trans. Actuarial Society of Am.*, 45:202–265, 1944.

[Güdesen, 1976] Güdesen, A. Quantitive analysis of preprocessing techniques for the recognition of handprinted characters. *Pattern Recognition*, 8:219–227, 1976.

[Guillemin, 1963] Guillemin, E.A. *Theory of Linear Physical Systems*. J. Wiley, New York, 1963.

[Gustafsson et al., 2001] Gustafsson, H., Nordholm, S.E., Claesson, I. Spectral subtraction using reduced delay convolution and adaptive averaging. *IEEE Trans. on Speech and Audio Processing*, 9:799–807, 2001.

[Hanaki et al., 1976] Hanaki, S.I., Temma, T., Yoshida, H. An on-line character recognition aimed at a substitution for a billing machine keyboard. *Pattern Recognition*, 8:63–71, 1976.

[Haralick et al., 1987] Haralick, R.M., Sternberg, S.R., Zhuang, X. Image analysis using mathematical morphology. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:532–550, 1987.

[Harris und Stephens, 1988] Harris, C., Stephens, M. A combined corner and edge detector. In *Proc. 4^{th} Alvey Vision Conference*, S. 147–151. Univ. of Manchester, 1988.

[Harris, 1978] Harris, F.J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE*, 66:51–83, 1978.

[Heers et al., 2001] Heers, J., Schnörr, C., Stiehl, H.S. Globally convergent iterative numerical schemes for nonlinear variational image smoothing on a multiprocessor machine. *IEEE Trans. on Image Processing*, 10(6):852–864, 2001.

[Heidemann, 2005] Heidemann, G. The long-range saliency of edge- and corner-based salient points. *IEEE Trans. on Image Processing*, 14:1701–1706, 2005.

[Hellwig, 1977] Hellwig, G. *Partial Differential Equations*. B.G. Teubner, Stuttgart, Germany, 1977.

[Henderson, 1906] Henderson, R. A practical interpolation formula with a theoretical introduction. *Trans. Actuarial Society of Am.*, 9(35):211–224, 1906.

[Herp, 1980] Herp, A. *Interaktive automatisierte Präzisionsvermessung von Stereo-Röntgenbildern zur*

Lokalisierungsdiagnostik von Hüftendoprothesen. Dissertation, Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, 1980.

[Herp et al., 1980] Herp, A., Niemann, H., Probst, K.J. Interactive evaluation of stereo x-ray images from hip joint protheses. In E.S. Gelsema, L.N. Kanal, Hg., *Pattern Recognition in Practice*, S. 245–258. North Holland, Amsterdam, 1980.

[Heygster, 1979] Heygster, G. Wirkung von Rangordnungsoperatoren im Frequenzbereich und auf die mittlere lokale Varianz von Bildern. *Informatik Fachberichte*, 20:87–93, 1979.

[Hirsch und Ehrlicher, 1995] Hirsch, H., Ehrlicher, C. Noise estimation techniques for robust speech recognition. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 153–156. Detroit, USA, 1995.

[Hladůvka und Gröller, 2001] Hladůvka, J., Gröller, E. Direction-driven shape-based interpolation of volume data. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, Hg., *Vision, Modeling, and Visualization 2001 (Proceedings of the International Workshop*, S. 113–120. Akademische Verlagsgesellschaft (Berlin, Germany), Stuttgart, Germany, 2001.

[Hodgson et al., 1985] Hodgson, R.M., Bailey, D.G., Naylor, M.J., Ng, A.L.M., McNeil, S.J. Properties, implementations and applications of rank order filters. *Image and Vision Computing*, 3:3–14, 1985.

[Hou und Andrews, 1978] Hou, H.S., Andrews, H.C. Cubic splines for image interpolation and digital filtering. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 26:508–517, 1978.

[Hu und Loizou, 2003a] Hu, Y., Loizou, P.C. A generalized subspace approach for enhancing speech corrupted by colored noise. *IEEE Trans. on Speech and Audio Processing*, 11:334–341, 2003a.

[Hu und Loizou, 2003b] Hu, Y., Loizou, P.C. A perceptually motivated approach to speech enhancement. *IEEE Trans. on Speech and Audio Processing*, 11:457–465, 2003b.

[Huang et al., 1979] Huang, T.S., Yang, G.J., Tang, G.Y. A fast two-dimensional median filtering algorithm. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 27:13–18, 1979.

[Hwang et al., 2000] Hwang, W.-J., Lin, R.-S., Hwang, W.-L., Wu, C.-K. Multiplication-free fast codeword search algorithm using Haar transform with squared-distance measure. *Pattern Recognition Letters*, 21:399–405, 2000.

[Hyonam et al., 1990] Hyonam, J., Haralick, R., Shapiro, L.G. Toward the automatic generation of mathematical morphology procedures using predicate logic. In *Proc. Third Int. Conf. on Computer Vision*, S. 156–165. IEEE Computer Society Press, Osaka, Japan, 1990.

[Ingram und Preston, 1970] Ingram, M., Preston, K. Automatic analysis of blood cells. *Scientific American*, 223:72–82, 1970.

[Itakura, 1975] Itakura, F. Minimum predicition residual principle applied to speech recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 23:67–72, 1975.

[Jacquin, 1993] Jacquin, A.E. Fractal image coding: A review. *Proceedings of the IEEE*, 81(10):1451–1465, 1993.

[Jähne, 2000] Jähne, B. Neighborhood operators. S. 273–345. Academic Press, London, UK, 2000.

[Jähne et al., 1999a] Jähne, B., Haußecker, H., Geißler, P., Hg. *Handbook of Computer Vision and Applications*, Bd. 1, 2, and 3. Academic Press, New York, USA, 1999a.

[Jähne et al., 1999b] Jähne, B., Scharr, H., Körkel, S. Principles of filter design. In [Jähne et al., 1999a], S. 125–151.

[Jaschul, 1982] Jaschul, J. *Adaption vorverarbeiteter Sprachsignale zum Erreichen der Sprecherunabhängigkeit automatischer Spracherkennungssysteme*. Dissertation, Fakultät für Elektrotechnik, TU München, München, 1982.

[Jayant, 1974] Jayant, N.S. Digital coding of speech waveforms, PCM, DPCM, and DM quantizers. In *Proc. IEEE*, Bd. 62, S. 611–632, 1974.

[Jayant, 1976] Jayant, N.S., Hg. *Waveform Quantization and Coding*. IEEE Press, New York, 1976.

[Jayant und Noll, 1984] Jayant, N.S., Noll, P. *Digital Coding of Waveforms*. Prentice Hall, Englewood Cliffs, N.J., 1984.

[Jiang und Mojon, 2001] Jiang, X., Mojon, D. Blood vessel detection in retinal images by shape-based multi-thresholding. In B. Radig, S. Florczyk, Hg., *Pattern Recognition. Proc. 23rd DAGM Symposium*, S. 38–44. (Springer LNCS 2191, Berling Heidelberg, ISBN 3-540-42596-9), München, Germany, 2001.

[Jungmann, 1984] Jungmann, B. Segmentierung mit morphologischen Operationen. In W. Kropatsch, Hg., *Musterkennung*, S. 77–83. Springer, Berlin, 1984.

[Junqua und Haton, 1996] Junqua, J.-C., Haton, J.-P. *Robustness in Automatic Speech Recognition*. Kluwer Acad. Publ., Boston, 1996.

[Kämmerer, 1989] Kämmerer, B. *Sprecheradaption und Reduktion der Sprecherabhängigkeit für die automatische Spracherkennung*. Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 1989.

[Kapur et al., 1985] Kapur, J.N., Sahoo, P.K., Wong, A.K.C. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285, 1985.

[Karup, 1899] Karup, J. über eine neue mechanische Ausgleichsmethode. In G. King, Hg., *Trans. of the Second International Actuarial Congress*, S. 31–77. Charles and Edwin Layton, London, 1899.

[Katsaggelos, 1991] Katsaggelos, A.K., Hg. *Digital Image Restoration*, Bd. 23 von *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1991.

[Kaufman und Tekalp, 1991] Kaufman, H., Tekalp, A.M. Survey of estimation techniques in image restoration. *IEEE Control Systems Magazine*, 11:16–24, 1991.

[Kaup et al., 2005] Kaup, A., Meisinger, K., Aach, T. Frequency selective signal extrapolation with applications to error concealment in image communication. *Int. Journal of Electronic Communication*, 59:147–156, 2005.

[Kittler und Illingworth, 1986] Kittler, J., Illingworth, J. Minimum cross error thresholding. *Pattern Recognition*, 19:41–47, 1986.

[Klemm, 1994] Klemm, H. Spektrale Subtraktion mit linearer und nichtlinearer Filterung zur Unterdrückung von "musical tones" bei hoher Sprachqualität. In *Proc. DAGA*, S. 1381–1384, Vol. C, 1994.

[Klette, 2001] Klette, R. Digital geometry – the birth of a new discipline. In L.S. Davis, Hg., *Foundations of Image Understanding*, Kap. 2. Kluwer Academic Publishers, Boston, 2001.

[Ko und Lee, 1991] Ko, S.J., Lee, Y.H. Center weighted median filters and their application to image enhancement. *IEEE Trans. Circuits and Systems*, 38:984–993, 1991.

[Kong, 2001] Kong, T.Y. Digital topology. In L.S. Davis, Hg., *Foundations of Image Understanding*, Kap. 3. Kluwer Academic Publishers, Boston, 2001.

[Kramer und Bruckner, 1975] Kramer, H.P., Bruckner, J.P. Iterations of a nonlinear transformation for enhancement of digital images. *Pattern Recognition*, 7:53–58, 1975.

[Kreifelts, 1977] Kreifelts, T. Skelettierung und Linienverfolgung in rasterdigitalisierten Linienstrukturen. In H.H. Nagel, Hg., *Digitale Bildverarbeitung*, Bd. 8 von *Informatik Fachberichte*, S. 223–231. Springer, Berlin, Heidelberg, New York, 1977.

[Kuan et al., 1987] Kuan, D.T., Sawchuk, A.A., Strand, T.C., Chavel, P. Adaptive restoration of images with speckle. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 35:373–383, 1987.

[Kubitschek, 1979] Kubitschek, H.M. Digitalisierung und Vorverarbeitung von Stromabläufen. Studienarbeit, Lehrstuhl für Mustererkennung (Informatik 5), Univ. Erlangen-Nürnberg, Erlangen, 1979.

[Kushner et al., 1989] Kushner, W., Goncharoff, V., Wu, C., Nguyen, V., Damoulakis, J. The effects of subtractive-type speech enhancement/noise reduction algorithms on parameter estimation for improved recognition and coding in high noise environments. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 211–214. Glasgow, 1989.

[Kwag et al., 2002] Kwag, H.K., Kim, S.H., Jeong, S.H., Lee, G.S. Efficient skew estimation and correction algorithm for document images. *Image and Vision Computing*, 20(1):25–35, 2002.

[Kwok und Tsang, 2004] Kwok, J.T., Tsang, I.W. The pre-image problem in kernel methods. *IEEE Trans. on Neural Networks*, 15:1517–1525, 2004.

[Lacroix, 1980] Lacroix, A. *Digitale Filter*. R. Oldenbourg, München, 1980.

[Lam et al., 1992] Lam, L., Lee, S.W., Suen, C.Y. Thinning methodologies – a comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:869–885, 1992.

[Lam et al., 1993] Lam, L., Lee, S.W., Suen, C.Y. A systematic evaluation of skeletonization algorithms. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 7:1203–1225, 1993.

[Lam und Suen, 1995] Lam, L., Suen, C.Y. An evaluation of parallel thinning algorithms for character recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:914–919, 1995.

[Lansdown und Schofield, 1995] Lansdown, J., Schofield, S. Expressive rendering: A review of non-photorealistic techniques. *IEEE Computer Graphics and Applications*, 15(3):29–37, 1995.

[Le Bouquin Jeannès et al., 2001] Le Bouquin Jeannès, R., Scalart, P., Faucon, G., Beaugeant, C. Combined noise and echo reduction in hands-free systems: A survey. *IEEE Trans. on Speech and Audio Processing*, 9:808–820, 2001.

[Ledley, 1964] Ledley, R.S. High-speed automatic analysis of biomedical pictures. *Science*, 146:216–223, 1964.

[Lee, 2001] Lee, Z. Thresholding implemented in the frequency domain. *IEEE Trans. on Image Processing*, 10(5):708–714, 2001.

[Lehmann et al., 1999] Lehmann, T.M., Grönner, C., Spitzer, K. Survey: Interpolation methods in medical image processing. *IEEE Trans. on Medical Imaging*, 18:1049–1075, 1999.

[Lehmann und Palm, 2001] Lehmann, T.M., Palm, C. Color line search for illumination estimation in real-world scenes. *Journ. Optical Society of America A*, 18:2679–2691, 2001.

[Lester und Arridge, 1999] Lester, H., Arridge, S.R. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32:129–149, 1999.

[Leung und Lam, 1998] Leung, C.K., Lam, F.K. Maximum segmented image information thresholding. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 60:57–76, 1998.

[Levi und Montanari, 1970] Levi, G., Montanari, U. A grey-weighted skeleton. *Information and Control*, 17:62–91, 1970.

[Li et al., 1996] Li, W., Haese-Coat, V., Ronsin, J. Using adaptive genetic algorithms in the design of morphological filters in textural image processing. In *Nonlinear Image Processing VII*, S. 24–35. SPIE, 1996.

[Lim und Lee, 1990] Lim, Y.W., Lee, S.U. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23(9):935–952, 1990.

[Lin und Coyle, 1990] Lin, J.-H., Coyle, E.J. Minimum mean absolute error estimation over the class of generalized stack filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 38(4):663–678, 1990.

[Linde et al., 1980] Linde, Y., Buzo, A., Gray, R.M. An algorithm for vector quantizer design. *IEEE Trans. Communication Theory*, 28:84–95, 1980.

[Ma und Wan, 2001] Ma, C.M., Wan, S.Y. Parallel thinning algorithms on 3d (18,6) binary images. *Computer Vision and Image Understanding*, 80(3):364–378, 2001.

[Maintz und Viergever, 1998] Maintz, J.B.A., Viergever, M.A. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[Makhoul et al., 1985] Makhoul, J., Roucos, S., Gish, H. Vector quantization. *Proc. IEEE*, 73:1551–1588, 1985.

[Manay und Yezzi, 2003] Manay, S., Yezzi, A. Anti-geometric diffusion for adaptive thresholding and fast segmentation. *IEEE Trans. on Image Processing*, 12:1310–1323, 2003.

[Maragos und Shafer, 1987a] Maragos, P.A., Shafer, R.W. Morphological filters – Part I: Their set-theoretic analysis and relations to linear shift-invariant filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 35(8):1153–1169, 1987a.

[Maragos und Shafer, 1987b] Maragos, P.A., Shafer, R.W. Morphological filters – Part II: Their relations to median, order-statistic, and stack filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 35(8):1170–1184, 1987b.

[Martin, 1977] Martin, T.B. One way to talk to computers. *IEEE Spectrum*, 14(5):35–39, 1977.

[Mason und Clemens, 1968] Mason, S.J., Clemens, J.K. Character recognition in an experimental reading machine for the blind. In P.A. Kolers, M. Eden, Hg., *Recognizing Patterns*, S. 155–167. MIT Press, Cambridge, MA, 1968.

[Max, 1960] Max, J. Quantizing for minimum distortion. *IRE Trans. Information Theory*, 6:7–12, 1960.

[McWilliams und Sloane, 1978] McWilliams, J., Sloane, N. *The Theory of Error Correcting Codes*. North Holland, Amsterdam, 1978.

[Meijering et al., 2001] Meijering, E.H.W., Niessen, W.J., Viergever, M.A. Quantitative evaluation of convolution-based methodes for medical image interpolation. *Medical Image Analysis*, 5(2):111–126, 2001.

[Middleton, 1960] Middleton, D. *An Introduction to Statistical Communication Theory*. McGraw Hill, New York, 1960.

[Mika et al., 1999] Mika, S., Schölkopf, B., Smola, A.J., Müller, K.-R., Scholz, M., Rätsch, G. Kernel PCA and de-noising in feature spaces. In M.S. Kearns, S.A. Solla, D.A. Cohn, Hg., *Advances in Neural Information Processing Systems*, S. 536–542. MIT Press, Cambridge, MA, USA, 1999.

[Mikolajczyk und Schmid, 2004] Mikolajczyk, K., Schmid, C. Scale and affine invariant interest point detectors. *Int. Journal of Computer Vision*, 60:63–86, 2004.

[Mikolajczyk und Schmid, 2005] Mikolajczyk, K., Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.

[Montesinos et al., 1998] Montesinos, P., Gouet, V., Deriche, R. Differential invariants for color images. In *Proc. Int. Conference on Pattern Recognition (ICPR)*. Brisbane, Australia, 1998.

[Morel und Solimini, 1995] Morel, J.-M., Solimini, S. *Variational Methods in Image Segmentation*. Birkhäuser, Boston, 1995.

[Morrin, 1974] Morrin, T.H. A black-white representation of a gray-scale picture. *IEEE Trans. on Computers*, 23:184–186, 1974.

[Morrin, 1976] Morrin, T.H. Chain-link compression of arbitrary black-white images. *Computer Graphics and Image Processing*, 5:172–189, 1976.

[Mukhopadhyay und Chanda, 2002] Mukhopadhyay, S., Chanda, B. An edge preserving noise smoothing technique using multiscale morphology. *Signal Processing*, 82:527–544, 2002.

[Mumford und Shah, 1989] Mumford, D., Shah, J. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure and Applied Mathematics*, 42:577–685, 1989.

[Muñoz et al., 2001] Muñoz, A., Blu, T., Unser, M. Least-squares image resizing using finite differences. *IEEE Trans. on Image Processing*, 10(9):1365–1378, 2001.

[Murtag und Starck, 2003] Murtag, F., Starck, J.L. Quantization from Bayes factors with application to multilevel thresholding. *Pattern Recognition Letters*, 24:2001–2007, 2003.

[Murthy und Udupa, 1974] Murthy, I.S.N., Udupa, K.J. A search algorithm for skeletonization of thick patterns. *Computer Graphics and Image Processing*, 3:247–259, 1974.

[Musse et al., 1999] Musse, O., Heitz, F., Armspach, J.-P. 3D deformable image matching using multiscale minimization of global energy functions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, S. 478–484. IEEE, Fort Collins, Colorado, USA, 1999.

[Nacken, 1996] Nacken, P. Chamfer metrics, the medial axis and mathematical morphology. *Journal of Mathematical Imaging and Vision*, 6(2/3):235–248, 1996.

[Nagel und Rosenfeld, 1977] Nagel, R.N., Rosenfeld, A. Computer detection of freehand forgeries. *IEEE Trans. on Computers*, 26:895–905, 1977.

[Niemann, 1973] Niemann, H. Fourier Transformation zweidimensionaler Signale. *VDI-Zeitschrift*, 115:134–138, 291–297, 1973.

[Niemann, 1974] Niemann, H. *Methoden der Mustererkennung*. Akademische Verlagsgesellschaft, Frankfurt, 1974.

[Niemann, 1981] Niemann, H. *Pattern Analysis*. Springer Series in Information Sciences 4. Springer, Berlin, Heidelberg, New York, 1981.

[Niemann, 1990] Niemann, H. *Pattern Analysis and Understanding*. Springer Series in Information Sciences 4. Springer, Berlin, 2. Aufl., 1990.

[Niemann und Wu, 1993] Niemann, H., Wu, J.K. Neural network adaptive image coding. *IEEE Trans. on Neural Networks*, 4:615–627, 1993.

[Nieminen et al., 1987] Nieminen, A., Heinonen, P., Neuvo, Y. A new class of detail preserving filters for image processing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:74–90, 1987.

[Ohlander et al., 1978] Ohlander, R., Price, K., Reddy, D.R. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8:313–333, 1978.

[Oppenheim und Schafer, 1975] Oppenheim, A.V., Schafer, R.W. *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1975.

[Oppenheim und Schafer, 1989] Oppenheim, A.V., Schafer, R.W. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ, 1989.

[Oppenheim et al., 1983] Oppenheim, A.V., Willsky, A.S., Young, I.T. *Systems and Signals*. Prentice Hall, Englewood Cliffs, NJ, USA, 1983.

[Ortigueira und Machado, 2003] Ortigueira, M.D., Machado, J.A.T. Editorial: Special Issue fractional signal processing and applications. *Signal Processing*, 83(11), 2003.

[Otsu, 1978] Otsu, N. Discriminant and least-squares threshold selection. In *Proc. 4. Int. Joint Conf. on Pattern Recognition*, S. 592–596. Kyoto, Japan, 1978.

[Otsu, 1979] Otsu, N. A threshold selection method form gray-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9:62–66, 1979.

[Ozaktas et al., 2001] Ozaktas, H.M., Zalevsky, Z., Kutay, M.A. *The Fractional Fourier Transform with Applications in Optics and Signal Processing*. J. Wiley, New York, 2001.

[Paley und Wiener, 1934] Paley, R.E.A.C., Wiener, N. The Fourier transform in the complex domain. *American Mathematical Society*, 1934.

[Patane und Russo, 2001] Patane, G., Russo, M. The enhanced LBG algorithm. *Neural Networks*, 14(9):1219–1237, 2001.

[Patane und Russo, 2002] Patane, G., Russo, M. Fully automatic clustering system. *IEEE Trans. on Neural Networks*, 13:1285–1298, 2002.

[Paton, 1970] Paton, K. Conic sections in chromosome analysis. *Pattern Recognition*, 2:39–51, 1970.

[Paulus et al., 1995] Paulus, D., Greiner, T., Knüvener, C.L. Watershed transformation of time series of medical thermal images. In *Proc. SPIE Conf. on Intelligent Robots and Computer Vision XIV: Algorithms, Techniques, Active Vision, and Meterials Handling*. SPIE Proceedings 2588, Philadelphia, 1995.

[Pease, 1968] Pease, M.C. An adaptation of the fast Fourier transform to parallel processing. *Journal of the Association for Comp. Machinery*, 15:252–264, 1968.

[Perez und Gonzalez, 1987] Perez, A., Gonzalez, R.C. An iterative thresholding algorithm for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:742–751, 1987.

[Perona und Malik, 1990] Perona, P., Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[Petrou und Kittler, 1991] Petrou, M., Kittler, J. Optimal edge detectors for ramp edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:483–491, 1991.

[Petrovski, 1955] Petrovski, I.G. *Vorlesungen über partielle Differentialgleichungen*. B.G. Teubner, Stuttgart, Germany, 1955.

[Pok und Liu, 2003] Pok, G., Liu, J.and Nair, A.S. Selective removal of impulse noise based on homogeneity level information. *IEEE Trans. on Image Processing*, 12:85–92, 2003.

[Prager, 1980] Prager, J.M. Extracting and labelling boundary segments in natural scenes. *IEEE Trans.*

on Pattern Analysis and Machine Intelligence, 2:16–27, 1980.

[Pratt, 1991] Pratt, W.K. *Digital Image Processing*. Wiley-Interscience, New York, 2. Aufl., 1991.

[Preston, 1971] Preston, K. Feature extraction by Golay hexagonal pattern transformations. *IEEE Trans. on Computers*, 20:1007–1014, 1971.

[Prewitt, 1970] Prewitt, J. Object enhancement and extraction. *Picture Processing and Psychopictorics*, S. 75–149, 1970.

[Pun, 1980] Pun, T. A new method for grey-level picture thresholding using the entropy of the histogram. *Signal Processing*, 2:223–237, 1980.

[Rabiner et al., 1975] Rabiner, L.R., Sambur, M.S., Schmidt, C.E. Applications of a nonlinear smoothing algorithm to speech processing. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 23:552–557, 1975.

[Ragnemalm, 1992] Ragnemalm, L. Fast erosion and dilation by contour processing and thresholding of distance maps. *Pattern Recognition Letters*, 13:161–166, 1992.

[Ramesh und Burrus, 1974] Ramesh, R.C., Burrus, C.S. Fast convolution using Fermat number transforms with applications to digital filtering. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 22(2):87–97, 1974.

[Rao, 1976] Rao, T.C.M. Feature extraction for fingerprint recognition. *Pattern Recognition*, 8:181–192, 1976.

[Reed und Truong, 1975] Reed, I.S., Truong, T.K. The use of finite fields to compute convolutions. *IEEE Trans. on Information Theory*, 21(2):208–213, 1975.

[Regel, 1982] Regel, P. A module for acoustic-phonetic transcription of fluently spoken German speech. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-30(3):440–450, 1982.

[Reggazoni und Teschioni, 1997] Reggazoni, C.S., Teschioni, A. A new approach to vector median filtering based on space filling curves. *IEEE Trans. on Image Processing*, 6:1024–1037, 1997.

[Ridler und Calvard, 1978] Ridler, T.W., Calvard, S. Picture thresholding using an iterative selection method. *IEEE Trans. on Systems, Man, and Cybernetics*, 8:629–623, 1978.

[Rieger, 1979] Rieger, B. Skelettierungsverfahren für die automatische Schreibererkennung. In J.P. Foith, Hg., *Angewandte Szenenanalyse*, Bd. 20 von *Informatik Fachberichte*, S. 168–179. Springer, Berlin, Heidelberg, New York, 1979.

[Rockett, 2005] Rockett, P.E. An improved rotation-invariant thinning algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1671–1674, 2005.

[Rohr, 1997] Rohr, K. On 3D differential operators for detecting point landmarks. *Image and Vision Computing*, 15(3):219–233, 1997.

[Rosenfeld, 1970] Rosenfeld, A. A nonlinear edge detection technique. In *Proc. IEEE*, Bd. 58, S. 814–816, 1970.

[Rosenfeld, 1975] Rosenfeld, A. A characterization of parallel thinning algorithms. *Information and Control*, 29:286–291, 1975.

[Rosenfeld, 1979] Rosenfeld, A. *Picture Languages*, Kap. 2. Academic Press, New York, 1979.

[Rosenfeld und Kak, 1976] Rosenfeld, A., Kak, A.C. *Digital Picture Processing*, Kap. 4. Academic Press, New York, 1976.

[Rosenfeld und Pfaltz, 1966] Rosenfeld, A., Pfaltz, J. Sequential operations in digital picture processing. *Journal of the Association for Comp. Machinery*, 13:471–494, 1966.

[Rosin, 2001] Rosin, P.L. Unimodal thresholding. *Pattern Recognition*, 34(11):2083–2096, 2001.

[Rubanov et al., 1998] Rubanov, N.S., Bovbel, E.I., Kukharchik, P.D., Bodrov, V.J. The modified number theoretic transform over the direct sum of finite fields to compute the linear convolution. *IEEE Trans. on Signal Processing*, 46:813–817, 1998.

[Saha und Udupa, 2001] Saha, P.K., Udupa, J.K. Optimum image thresholding via class uncertainty and region homogeneity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:689–706, 2001.

[Sahoo et al., 1988] Sahoo, P.K., Soltani, S., Wong, A.K.C., Chen, Y.C. A survey of thresholding tech-

niques. *Computer Vision, Graphics, and Image Processing*, 41:233–260, 1988.

[Salisbury et al., 1994] Salisbury, M.P., Anderson, S.E., Barzel, R. Interactive pen-and-ink illustration. In A. Glassner, Hg., *Proc. SIGGRAPH*, S. 101–108. Orlando, FL, USA, 1994.

[Schafer und Rabiner, 1975] Schafer, R.W., Rabiner, L.R. Parametric representation of speech. In D.R. Reddy, Hg., *Speech Recognition*, S. 99–150. Academic Press, New York, 1975.

[Schless, 1999] Schless, V. *Automatische Erkennung von gestörten Sprachsignalen*. Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 1999.

[Schmid et al., 2000] Schmid, C., Mohr, R., Bauckhage, C. Evaluation of interest point detectors. *Int. Journal of Computer Vision*, 37:151–172, 2000.

[Schnörr, 1998] Schnörr, C. A study of a convex variational diffusion approach for image segmentation and feature extraction. *Journal of Mathematical Imaging and Vision*, 8:271–292, 1998.

[Schukat-Talamazzini et al., 1993] Schukat-Talamazzini, E.G., Bielecki, M., Niemann, H., Kuhn, T., Rieck, S. A non-metrical space search algorithm for fast Gaussian vector quantization. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. II–688 – II–691. Minneapolis, MN, 1993.

[Schürmann, 1974] Schürmann, J. Bildvorverarbeitung für die automatische Zeichenerkennung. *Wiss. Ber. AEG-Telefunken*, 47(3, 4):90–99, 1974.

[Schürmann, 1978] Schürmann, J. A multifont word recognition system for postal address reading. *IEEE Trans. on Computers*, 27:721–732, 1978.

[Schüßler, 1973] Schüßler, H.W. *Digitale Systeme zur Signalverarbeitung*. Springer, Berlin, 1973.

[Schüßler, 1992] Schüßler, H.W. *Digitale Signalverarbeitung*. Springer, Berlin, 3. Aufl., 1992.

[Şenel et al., 2002] Şenel, H.G., Peters, R.A., Dawant, B. Topological median filters. *IEEE Trans. on Image Processing*, 11:89–104, 2002.

[Serra, 1982] Serra, J. *Image Analysis and Mathematical Morphology, Vol. 1*. Academic Press, London, 1982.

[Serra, 1986] Serra, J. Introduction to mathematical morphology. *Computer Vision, Graphics, and Image Processing*, 35:283–305, 1986.

[Serra, 1988] Serra, J. *Image Analysis and Mathematical Morphology, Vol. 2*. Academic Press, London, 1988.

[Serra und Soille, 1994] Serra, J., Soille, P., Hg. *Mathematical Morphology and its Applications to Image Processing*, Bd. 2 von *Computational Imaging and Vision*. Kluwer Academic Publishers, Dordrecht, 1994.

[Shaick et al., 2000] Shaick, B.-Z., Riedel, L., Yaroslavsky, L. A hybrid transform method for image denoising. In *Proc. European Signal Processing Conference*, S. 2449–2452. Tampere, Finland, 2000.

[Shen und Castan, 1986] Shen, J., Castan, S. An optimal linear operator for edge detection. In *Proc. Computer Vision, Graphics, and Image Processing*, S. 109–114. Miami, FL, 1986.

[Shen, 1992] Shen, S.C.J. An optimal linear operator for step edge detection. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 54:112–133, 1992.

[Shen und Zhao, 1990] Shen, S.C.J., Zhao, J. New edge detection methods based on exponential filter. In *Proc. Int. Conference on Pattern Recognition (ICPR)*. Atlantic City, 1990.

[Shih et al., 2003] Shih, A.C.-C., Liao, H.-Y.M., Lu, C.-S. A new iterated two-band diffusion equation: Theory and its application. *IEEE Trans. on Image Processing*, 12:466–476, 2003.

[Silverman und Dixon, 1974] Silverman, H.F., Dixon, N.R. A parametrically controlled spectral analysis system for speech. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 22:362–381, 1974.

[Sklansky und Nahin, 1972] Sklansky, J., Nahin, P.J. A parallel mechanism for describing silhouettes. *IEEE Trans. on Computers*, 21:1233–1239, 1972.

[Smith und Brady, 1997] Smith, S.M., Brady, J.M. SUSAN – A new approach to low level image processing. *Int. Journal of Computer Vision*, 23:45–78, 1997.

[Soille, 1999a] Soille, P. *Morphological Image Analysis – Priciples and Applications*. Springer, Berlin, Germany, 1999a.

[Soille, 1999b] Soille, P. Morphological operators. In [Jähne et al., 1999a], S. 627–681.

[Soille, 2002] Soille, P. On morphological operators based on rank filters. *Pattern Recognition*, 35:527–535, 2002.

[Soille et al., 1996] Soille, P., Breen, E., Jones, R. Recursive implementations of erosions and dilations along discrete lines at arbitrary angles. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:562–567, 1996.

[Stefanelli und Rosenfeld, 1971] Stefanelli, R., Rosenfeld, A. Some parallel thinning algorithms for digital pictures. *Journal of the Association for Comp. Machinery*, 18:225–264, 1971.

[Sternberg, 1986] Sternberg, S.R. Grayscale morphology. *Computer Vision, Graphics, and Image Processing*, 35:333–355, 1986.

[Tang et al., 1995] Tang, K., Astola, J., Neuvo, Y. Nonlinear multivariate image filtering technique. *IEEE Trans. on Image Processing*, 6:788–798, 1995.

[Tekalp et al., 1989] Tekalp, A.M., Kaufman, H., Woods, J.W. Edge-adaptive Kalman filtering for image restoration with ringing suppression. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37:892–899, 1989.

[Temes und Mitra, 1973] Temes, G.C., Mitra, S.K. *Modern Filter Theory and Design*. J. Wiley, New York, 1973.

[Thévenaz et al., 2000a] Thévenaz, P., Blu, T., Unser, M. Image interpolation and resampling. In I.N. Bankman, Hg., *Handbook of Medical Imaging, Processing and Analysis*, S. 393–420. Academic Press, San Diego, CA, USA, 2000a.

[Thévenaz et al., 2000b] Thévenaz, P., Blu, T., Unser, M. Interpolation revisited. *IEEE Trans. on Medical Imaging*, 19:739–758, 2000b.

[Tian et al., 2001] Tian, Q., Sebe, M., Lew, M.S., Loupias, E., Huang, T.S. Image retrieval using wavelet based salient points. *J. Electron. Imaging*, 10:835–849, 2001.

[Triendl, 1970] Triendl, E. Skeletonization of noisy handdrawn symbols using parallel operations. *Pattern Recognition*, 2:215–226, 1970.

[Tsai et al., 2001] Tsai, C., Yezzi Jr., A., Willsky, A.S. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. on Image Processing*, 10:1169–1186, 2001.

[Tsai, 1995] Tsai, D.M. A fast thresholding selection procedure for multimodal and unimodal histograms. *Pattern Recognition Letters*, 16:653–666, 1995.

[Tsai, 1985] Tsai, W.H. Moment-preserving thresholding. *Computer Vision, Graphics, and Image Processing*, 29:377–394, 1985.

[Tschumperlé und Deriche, 2002] Tschumperlé, D., Deriche, R. Diffusion PDEs on vector-valued images. *IEEE Signal Processing Magazine*, 19(5):16–25, 2002.

[Unser, 1999] Unser, M. Splines: A perfect fit for signal and image processing. *IEEE Trans. on Signal Processing*, 16(6):22–38, 1999.

[Unser et al., 1991] Unser, M., Aldroubi, A., Eden, M. Fast B-spline transform for continuous image representation and interpolation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:277–285, 1991.

[Unser et al., 1993a] Unser, M., Aldroubi, A., Eden, M. B-spline signal processing: Part I – theory. *IEEE Trans. on Signal Processing*, 41:821–833, 1993a.

[Unser et al., 1993b] Unser, M., Aldroubi, A., Eden, M. B-spline signal processing: Part II – efficient design and applications. *IEEE Trans. on Signal Processing*, 41:834–848, 1993b.

[Unser et al., 1995] Unser, M., Thévenaz, P., Yaroslavsky, L. Convolution-based interpolation for fast, high-quality rotation of images. *IEEE Trans. on Image Processing*, 4:1371–1381, 1995.

[van den Boomgaard und van Balen, 1992] van den Boomgaard, R., van Balen, R. Methods for fast morphological image transforms using bitmapped binary images. *Computer Vision, Graphics,*

and Image Processing: Graphical Models and Image Processing, 54(3):252–258, 1992.

[van Herk, 1992] van Herk, M. A fast algorithm for local minimum and maximum filters on rectangular and hexagonal kernels. *Pattern Recognition Letters*, 13:517–521, 1992.

[Van Voorhis, 1976] Van Voorhis, D.C. An extended run-length encoder and decoder for compression of black/white images. *IEEE Trans. on Information Theory*, 22:190–199, 1976.

[Vardavoulia et al., 2001] Vardavoulia, M.I., Andreadis, I., Tsaldis, Ph. A new vector median filter for colour image processing. *Pattern Recognition Letters*, 22:675–689, 2001.

[Vermaak et al., 2002] Vermaak, J., Andrieu, C., Doucet, A., Godsill, S.J. Particle methods for Bayesian modeling and enhancement of speech signals. *IEEE Trans. on Speech and Audio Processing*, 10:173–185, 2002.

[Vincent, 1991] Vincent, L. Morphological transformations of binary images with arbitrary structuring elements. *Signal Processing*, 22(1):3–23, 1991.

[Vincent, 1993] Vincent, L. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Trans. on Image Processing*, 2(2):176–201, 1993.

[Vincent und Soille, 1991] Vincent, L., Soille, B. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:583–598, 1991.

[Vogt, 1989] Vogt, R.C. *Automatic Generation of Morphological Set Recognition Algorithms*. Springer Series in Perception Engineering. Springer, Berlin, Heidelberg, 1989.

[Wakita, 1977] Wakita, H. Normalization of vowels by vocal-tract length and its application to vowel identification. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 25:183–192, 1977.

[Wakita, 1979] Wakita, H. Estimation of vocal-tract shapes from acoustical analysis of the speech wave; the state of the art. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 27:281–285, 1979.

[Wang und Bai, 2003] Wang, L., Bai, J. Threshold selection by clustering gray levels of boundary. *Pattern Recognition Letters*, 24:1983–1999, 2003.

[Wang und Zhang, 1999] Wang, Z., Zhang, D. Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Trans. on Circuits and Systems*, 46:78–80, 1999.

[Weickert, 1998] Weickert, J. *Anisotropic Diffusion in Image Processing*. B.G. Teubner, Stuttgart, 1998.

[Weickert, 1999a] Weickert, J. Coherence enhancing diffusion filtering. *Int. Journal of Computer Vision*, 1999a.

[Weickert, 1999b] Weickert, J. Nonlinear diffusion filtering. In [Jähne et al., 1999a], S. 423–448.

[Weickert et al., 1998] Weickert, J., ter Haar Romeney, B., Viergever, M. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. on Image Processing*, 7:398–410, 1998.

[Welling et al., 2002] Welling, L., Ney, H., Kanthak, S. Speaker adaptive modelling by vocal tract normalization. *IEEE Trans. on Speech and Audio Processing*, 10:415–426, 2002.

[Wendland, 1995] Wendland, H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.

[Wendt et al., 1986] Wendt, P.D., Coyle, E.J., Gallagher Jr., N.C. Stack filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 34:898–911, 1986.

[Weszka, 1978] Weszka, J.S. A survey of threshold selection techniques. *Computer Graphics and Image Processing*, 7:259–265, 1978.

[Weszka et al., 1974] Weszka, J.S., Nagel, R.N., Rosenfeld, A. A threshold selection technique. *IEEE Trans. on Computers*, 23:1322–1326, 1974.

[Winkenbach und Salesin, 1994] Winkenbach, G., Salesin, D.H. Computer generated pen-and-ink illustration. In A. Glassner, Hg., *Proc. SIGGRAPH*, S. 91–100. Orlando, FL, USA, 1994.

[Winkler, 1977] Winkler, G. *Stochastische Systeme, Analyse and Synthese*. Akademische Verlagsgesellschaft, Wiesbaden, 1977.

[Woods und Ingle, 1981] Woods, J.W., Ingle, V.K. Kalman filtering in two dimensions – further results. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 29:188–197, 1981.

[Woods und Radewan, 1977] Woods, J.W., Radewan, C.H. Kalman filter in two dimensions. *IEEE Trans. on Information Theory*, 23:473–482, 1977.

[Xu et al., 2000] Xu, F., Liu, H. Wang, G., Alford, B.A. Comparison of adaptive linear interpolation and conventional linear interpolation for digital radiography systems. *Journal of Electronic Imaging*, 9(1):22–31, 2000.

[Yang und Li, 1995] Yang, J., Li, X. Boundary detection using mathematical morphology. *Pattern Recognition Letters*, 16:1277–1286, 1995.

[Yang und Tsai, 1998] Yang, J.C., Tsai, W.-H. Suppression of Moiré patterns in scanned halftone images. *Signal Processing*, 70:23–42, 1998.

[Yang und Tsai, 2000] Yang, J.C., Tsai, W.-H. Document image segmentation and quality improvement by Moiré pattern analysis. *Signal Processing: Image Communication*, 15:781–797, 2000.

[Yaroslavski, 1997] Yaroslavski, L. Local adaptive filtering in transform domain for image restoration, enhancement and target location. In E. Wenger, L.I. Dimitrov, Hg., *Proc. SPIE 6-th Int. Workshop on Digital Image Processing and Computer Graphics*, S. SPIE Vol. 3346, 1–17. Vienna, Austria, 1997.

[Yin et al., 1996] Yin, L., Yang, R., Gabbouj, M., Neuvo, Y. Weighted median filters: A tutorial. *IEEE Trans. Circuits and Systems II*, 43:157–192, 1996.

[Young und van Vliet, 1995] Young, I.T., van Vliet, L.J. Recursive implementation of the Gaussian filter. *Signal Processing*, 44:139–151, 1995.

[Yu und Acton, 2002] Yu, Y., Acton, S.T. Speckle reducing anisotropic diffusion. *IEEE Trans. on Image Processing*, 11:1260–1270, 2002.

[Zheng et al., 1993] Zheng, J., Valavanis, K.P., Gauch, J.M. Noise removal from colour images. *J. Intelligent Robotic Systems*, 7:257–285, 1993.

[Zhou et al., 1995] Zhou, R.W., Quek, C., Ng, G.S. A novel single-pass thinning algorithm and an effective set of performance criteria. *Pattern Recognition Letters*, 16:1267–1275, 1995.

[Zygmund, 1968] Zygmund, A. *Trigonometric Series*, Bd. 2. Cambridge University Press, Cambridge, 1968.

Kapitel 3

Merkmale (VK.2.3.3, 13.04.2004)

Man kann die Klassifikation von Mustern allgemein als eine Abbildung auffassen, die einem Muster $\boldsymbol{f}(\boldsymbol{x})$ eine ganze Zahl $\kappa \in \{0, 1, \ldots, k\}$, nämlich die Nummer der Klasse, zuordnet. Erfahrungsgemäß ist es aber für praktisch interessante Fälle oft nicht möglich, eine derartige Abbildung „in einem Schritt“ zu finden. Daher wird das Ziel der Klassifikation in mehrere Unterziele oder Teilschritte zerlegt, wobei man jeden Teilschritt so wählen sollte, dass er zum einen ein lösbares Teilproblem enthält und zum anderen für die Erreichung des Gesamtzieles förderlich ist. Leider ist man bei der Zerlegung des Problems der Klassifikation in Teilprobleme weitgehend auf heuristische, intuitive und empirische Gesichtspunkte angewiesen, d. h. es gibt keinen Algorithmus, der für einen vorgegebenen Problemkreis und eine vorgegebene Definition der Leistungsfähigkeit eines Klassifikationssystems eine gute oder gar optimale Zerlegung liefert. Es hat sich aber immer wieder gezeigt, dass die in Bild 1.4.1, S. 26, gezeigte Zerlegung in die Teilschritte der Vorverarbeitung, Merkmalgewinnung und Klassifikation praktisch brauchbar und erfolgreich ist. Natürlich gibt es von dieser Standardzerlegung Abweichungen, Modifikationen und Verfeinerungen, aber die Grundstruktur bleibt stets erkennbar.

Wenn man sich für ein Vorgehen gemäß dieser Grundstruktur entschieden hat, so gibt es zur Durchführung jedes Teilschrittes oder zur Lösung jedes Teilproblems wiederum zahlreiche Ansätze, wofür das vorige Kapitel ein Beispiel ist. In diesem Kapitel werden Ansätze zur Gewinnung von Merkmalen erörtert, und zwar werden die folgenden Themen behandelt:

1. Anliegen und allgemeine Ansätze – eine Verdeutlichung des Problems und der prinzipiellen Lösungsmöglichkeiten.
2. Entwicklung nach einer Orthogonalbasis – die Verwendung der Koeffizienten einer Reihenentwicklung als Merkmale.
3. Wavelet Transformation – die Verwendung einer speziellen Reihenentwicklung.
4. Filterbänke – (digitale) Filter zur Gewinnung von Merkmalen.
5. Andere heuristische Verfahren – einige Ergänzungen zu den obigen Themen.
6. Merkmale für die Texturerkennung – spezielle Merkmale für dieses spezielle Problem.
7. Merkmale für die Spracherkennung – erfahrungsgemäß sind hier Techniken zu nutzen, die sich von denen für bildhafte Muster deutlich unterscheiden.
8. Merkmale für die Objekterkennung – hier stehen Merkmale für zwei– und dreidimensionale Objekte im Vordergrund.
9. Analytische Verfahren – Methoden zur Gewinnung von Merkmalvektoren, die ein vorgegebenes Kriterium optimieren.

10. Merkmalbewertung und –auswahl – Verfahren zur Ermittlung einer möglichst guten Untermenge von Merkmalen aus einer Menge vorgegebener Merkmale.
11. Symbole – Gewinnung von Merkmalen, die durch Symbole gekennzeichnet werden.

Es wird noch an die Bemerkung in Kapitel 2 erinnert, dass unter Umständen die Grenze zwischen Vorverarbeitung und Merkmalgewinnung unscharf ist.

3.1 Anliegen und allgemeine Ansätze (VA.1.2.2, 15.11.2005)

Die Einführung der Stufe **„Merkmale“** zwischen Vorverarbeitung und Klassifikation in Bild 1.4.1, S. 26, beruht darauf, dass eine direkte Klassifikation der Abtastwerte $\boldsymbol{f}$ eines Musters $\boldsymbol{f}(\boldsymbol{x})$ vielfach unmöglich oder unzweckmäßig erscheint. Sie erscheint vielfach unmöglich wegen der großen Zahl von Abtastwerten, die dann vom Klassifikator verarbeitet werden müssten. So werden z. B. Schriftzeichen mit einer anfänglichen Auflösung bis zu $40 \times 30 = 1200$ Abtastwerten dargestellt; ein isoliert gesprochenes Wort von 1 s Dauer ergibt bei einer Abtastfrequenz von 10 kHz bereits 10.000 Abtastwerte; und ein mit Fernsehqualität aufgenommener Fingerabdruck liefert etwa $512 \times 512 \simeq 260.000$ Abtastwerte. Ein Zweck der Merkmalsgewinnung ist also die *Reduktion der Datenmenge*. Die direkte Klassifikation der Abtastwerte erscheint vielfach unzweckmäßig, da die vollständige Darstellung des Musters (im Sinne von Satz 2.1, S. 65) für die Klassifikation weniger wichtig ist als die Herausarbeitung der „trennscharfen“ Information, welche die Unterscheidung von Mustern verschiedener Klassen erlaubt. Zum Beispiel ist für die Unterscheidung der großen Druckbuchstaben O und Q vor allem der rechte untere Teil wichtig. Merkmale sollten also möglichst die für die Klassifikation charakteristischen Eigenschaften der Muster enthalten, aber auch nicht mehr. Ein weiterer Zweck der Merkmalsgewinnung ist also die *Konzentration auf die für die Klassifikation wichtige Information*.

Man wird bestrebt sein, solche Merkmale zu finden, welche die „Güte“ des Gesamtsystems maximieren. Bei der Beurteilung der Güte sollten unter anderem die Kosten des Systems, die Verarbeitungsgeschwindigkeit und die Fehlerwahrscheinlichkeit bei der Klassifikation von Mustern berücksichtigt werden, bzw. die in (1.8.1), S. 49, genannten Größen. Bisher gibt es keine Algorithmen, mit denen solche Merkmale systematisch erzeugt werden können. Daher werden die an sich wünschenswerten Anforderungen an Merkmale reduziert, wobei vor allem zwei Einschränkungen vorgenommen werden:

1. Als Gütekriterium werden nur die Fehlerwahrscheinlichkeit und die Zahl der Merkmale betrachtet. Statt der Fehlerwahrscheinlichkeit wird zudem oft ein mathematisch einfacher zu behandelndes Kriterium gewählt.
2. Es wird nicht das Gesamtsystem betrachtet, sondern nur der Modul Merkmalsgewinnung, d. h. die Merkmale werden weitgehend unabhängig von den Vorverarbeitungsoperationen und dem Klassifikator ermittelt.

Mit diesen beiden Vereinfachungen wird in praktisch allen Ansätzen zur Merkmalsgewinnung gearbeitet; eine Ausnahme enthält Abschnitt 3.8.4.

Es lassen sich zwei grundsätzliche Typen von Merkmalen unterscheiden, nämlich die durch reelle Zahlen und die durch Symbole (oder auch nominale Merkmale) gekennzeichneten. Im ersten Falle wird jedem Muster ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x})$ oder dessen Abtastwerten ${}^{\varrho}\boldsymbol{f}$ mit einer Transformation T_{r} ein **Merkmalsvektor**

$$ {}^{\varrho}\boldsymbol{c} = T_{\mathrm{r}}\{{}^{\varrho}\boldsymbol{f}\}\,, \quad \varrho = 1, 2, \ldots \tag{3.1.1} $$

zugeordnet. Im Folgenden wird stets angenommen, dass der Merkmalsvektor ein Spaltenvektor

$$ {}^{\varrho}\boldsymbol{c} = ({}^{\varrho}c_1, {}^{\varrho}c_2, \ldots, {}^{\varrho}c_\nu, \ldots, {}^{\varrho}c_n)^{\mathsf{T}}\,, \quad {}^{\varrho}c_\nu \in \mathbb{R} \tag{3.1.2} $$

mit n reellwertigen Komponenten ${}^{\varrho}c_\nu$ ist. Der Index $^{\mathsf{T}}$ in (3.1.2) kennzeichnet die Transponierung des Vektors. Falls die Bezeichnung individueller Muster und Merkmalsvektoren unnötig

ist, wird auch der links oben stehende Index ϱ fortgelassen. Merkmalsvektoren gemäß (3.1.1), (3.1.2) sind geeignet als Eingangsgrößen für die im Kapitel 4 zu behandelnden numerischen Klassifikatoren. Ein Beispiel für derartige Merkmale ist die Verwendung des Betrages der ersten n Koeffizienten der DFT gemäß (2.3.28), S. 93.

In (3.1.1) kommt zum Ausdruck, dass für jede der k Musterklassen der *gleiche* Merkmalsvektor verwendet wird. In Abschnitt 4.1.7 wird gezeigt, dass die Klassifikation von Mustern mit dem entscheidungstheoretischen Ansatz auch dann möglich ist, wenn **klassenspezifische Merkmale** verwendet werden. In diesem Falle hat man k Transformationen $T_{\mathrm{r}}^{(\kappa)}$ zur Berechnung von k klassenspezifischen Merkmalsvektoren ${}^{\varrho}\boldsymbol{c}^{(\kappa)}$

$$ {}^{\varrho}\boldsymbol{c}^{(\kappa)} = T_{\mathrm{r}}^{(\kappa)}\{{}^{\varrho}\boldsymbol{f}\}\ , \quad \varrho = 1, 2, \dots\ , \quad \kappa = 1, \dots, k\ . \tag{3.1.3} $$

Damit hat man im Prinzip die Möglichkeit, für jede Klasse die jeweils am besten geeigneten Merkmale zu verwenden. Hierfür gibt es noch kaum konkrete Hinweise, wie diese Merkmale zu finden sind. Die Diskussion in diesem Kapitel wird sich daher auf die Ermittlung eines Merkmalsvektors gemäß (3.1.1) konzentrieren.

Die Transformationen in (3.1.1) und (3.1.3) wirken auf das gesamte Muster. Der Wert einer Komponente c_ν des Merkmalsvektors $\boldsymbol{c}$ bzw. so eines **globalen Merkmals** c_ν hängt damit vom gesamten Muster ab und ändert sich demzufolge, wenn sich nur ein kleiner Bereich des Musters, z. B. in Folge von Störungen oder Verdeckungen, verändert. Um dieses zu vermeiden werden stattdessen auch **lokale Merkmale** verwendet, deren Wert nur von einer kleinen Nachbarschaft des Musters abhängt. Änderungen im Muster außerhalb dieser Nachbarschaft haben dann keinen Einfluss auf dieses Merkmals, sondern betreffen nur die Merkmale, deren Nachbarschaften von der Änderung betroffen sind. Formal werden lokale Merkmale durch Einführung einer Fensterfunktion $w(\boldsymbol{x})$ gebildet, die aus dem Muster eine Teilmenge von Werten „ausschneidet“. Beispiele für solche Funktionen wurden in (2.5.43), S. 131, angegeben. Die Fensterfunktion wird an die Stelle $\boldsymbol{x}_m$ gelegt und bewichtet die Funktionswerte des Musters. An der Position $\boldsymbol{x}_m$ kann lediglich ein Merkmal $c_\nu(\boldsymbol{x}_m)$ oder ein Vektor von Werten $\boldsymbol{c}(\boldsymbol{x}_m) = \boldsymbol{c}_m$ berechnet werden. Der m-te lokale Merkmalsvektor ergibt sich aus

$$ {}^{\varrho}\boldsymbol{c}_m = T_{\mathrm{r}}\{w(\boldsymbol{x} - \boldsymbol{x}_m){}^{\varrho}\boldsymbol{f}\}\ , \quad \varrho = 1, 2, \dots\ . \tag{3.1.4} $$

Im Folgenden geht aus dem Kontext hervor, welcher Typ von Merkmalen jeweils berechnet wird. Die Positionen $\boldsymbol{x}_m$, an denen lokale Merkmale berechnet werden, können auf einem regelmäßigen Gitter liegen wie in Abschnitt 3.7.2, oder an markanten Punkten.

Im zweiten Falle wird jedem Muster ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x})$ oder dessen Abtastwerten ${}^{\varrho}\boldsymbol{f}$ mit einer Transformation T_{s} eine **Symbolkette**

$$ {}^{\varrho}v = T_{\mathrm{s}}\{{}^{\varrho}\boldsymbol{f}\} \tag{3.1.5} $$

zugeordnet. Die Symbolkette

$$ {}^{\varrho}v = {}^{\varrho}v_1\,{}^{\varrho}v_2 \dots {}^{\varrho}v_{n(\varrho)}\ , \quad {}^{\varrho}v_j \in V_T \tag{3.1.6} $$

hat die Länge $n(\varrho)$, und jede Komponente ${}^{\varrho}v_j$ ist ein Element aus einer endlichen Menge V_T von einfacheren Bestandteilen des Musters. Diese Menge wird auch terminales Alphabet genannt; sie kann bei Bedarf außer den einfacheren Bestandteilen auch Relationen zwischen diesen enthalten. Symbolketten gemäß (3.1.5), (3.1.6) sind als Eingangsgrößen für die z. B. in [Fu, 1974, Fu, 1982, Gonzalez und Thomason, 1978, Niemann, 1974, Niemann, 1983] behandelten *syntaktischen Klassifikatoren* geeignet, die hier nicht weiter beschrieben werden; lediglich Abschnitt 4.6.3 gibt ein Beispiel für einen Klassifikator für symbolische bzw. *nominale*

Merkmale und mit (4.6.23), S. 406 wird die Berechnung von Abständen zwischen Merkmalsvektoren auf die von Symbolketten verallgemeinert. Symbolische Merkmale spielen auch bei numerischen Klassifikatoren eine Rolle und werden daher hier aufgenommen. Ein Beispiel für derartige Merkmale ist die Beschreibung einer Konturlinie durch Linienelemente wie „stark konvex", „schwach konvex", „stark konkav" oder „gerade". Die Relation zwischen den Merkmalen ist hier einfach die Aneinanderreihung und braucht daher nicht ausdrücklich angegeben zu werden. Allgemeinere Relationen sind z. B. „enthalten in", „unter", „links", „rechts über".

Eine Symbolkette ${}^{\varrho}v$ lässt sich stets in einen Merkmalsvektor ${}^{\varrho}\boldsymbol{c}$ umformen, indem man die Elemente von V_T in irgendeiner Weise durch Zahlen codiert. Eine naheliegende Methode besteht darin, einen binären Vektor aufzubauen, in dem eine 1 in einer Komponente anzeigt, dass z. B. das Merkmal „gerades Linienelement" vorhanden ist, und eine 0 anzeigt, dass es nicht vorhanden ist. Trotzdem sind beide Ansätze nicht ohne weiteres als identisch zu betrachten, da diese Umformung zu einer Verletzung von Postulat 3, S. 20, führen kann, d. h. man muss dann nach besseren Merkmalen suchen.

Zur Gewinnung von Merkmalen werden hier zwei grundlegende Ansätze unterschieden:

1. Die heuristische Methode, bei der man versucht, Merkmale aufgrund von Intuition, Phantasie und Erfahrung zu finden. Dazu gehören die Verfahren in Abschnitt 3.2 – Abschnitt 3.6 sowie in Abschnitt 3.10.
2. Die analytische Methode, bei der man versucht, in bestimmtem Sinne optimale Merkmale systematisch abzuleiten. Dazu gehören die Verfahren in Abschnitt 3.8

Zur Lösung praktisch interessanter Probleme wird oft eine – ebenfalls heuristisch gefundene – Kombination beider Ansätze verwendet. Bei der heuristischen Vorgehensweise erhält man eine Menge von Merkmalen, deren Eignung für das jeweilige Klassifikationsproblem recht unterschiedlich sein kann. Da der Aufwand zur Durchführung einer Klassifikation mit der Zahl der Merkmale anwächst, wird man bestrebt sein, die weniger geeigneten Merkmale zu eliminieren. Als weiterer wichtiger Ansatz kommt daher als Ergänzung hinzu:

3. Die Bewertung einer vorgegebenen Menge von Merkmalen und die Auswahl einer möglichst guten Untermenge in Abschnitt 3.9.

Die Merkmalsgewinnung mit der analytischen Methode liefert die n besten Merkmale (im Sinne des vorgegebenen Gütekriteriums) in einem Schritt. Die heuristische Methode ergibt zunächst einmal eine Menge von Merkmalen, über deren Qualität entweder nur Mutmaßungen möglich sind oder experimentelle Hinweise der Literatur für vergleichbare Probleme entnommen werden können. Durch den zusätzlichen Schritt der Bewertung und Auswahl von Merkmalen ist es möglich, aus der heuristisch gefundenen Merkmalsmenge eine sehr gute, wenn auch i. Allg. nicht die optimale, Untermenge zu ermitteln.

Sowohl die heuristische als auch die analytische Methode werden zur Gewinnung von Transformationen T_{r} gemäß (3.1.1) eingesetzt; dazu kommt gegebenenfalls eine Bewertung und Auswahl von Merkmalen. Zur Gewinnung von Transformationen T_{s} gemäß (3.1.5) wird bisher nur die heuristische Methode verwendet, da analytische Ansätze und Bewertungs– und Auswahlverfahren für diese Merkmale zur Zeit fehlen.

3.2 Orthogonale Reihenentwicklung (VA.1.2.2, 07.02.2004)

3.2.1 Allgemeine Beziehungen

Eine naheliegende Methode zur Merkmalsgewinnung besteht in der Entwicklung des Musters $\boldsymbol{f}(\boldsymbol{x})$ nach einem orthonormalen Funktionensystem $\varphi_\nu(\boldsymbol{x})$ oder des Vektors $\boldsymbol{f}$ von Abtastwerten nach orthogonalen Basisvektoren $\boldsymbol{\varphi}_\nu$. Im Prinzip kommt es dabei weniger auf die Orthogonalität der Basisvektoren als auf die *Eindeutigkeit* der Entwicklungskoeffizienten an. Allerdings sind die numerischen Rechnungen bei orthogonalen Basisvektoren besonders einfach. Die *Heuristik* liegt in der Annahme, dass die Entwicklungskoeffizienten als Merkmale für die Klassifikation von Mustern geeignet sind.

Definition 3.1 *Eine Menge von Vektoren* $\widetilde{\varphi} = \{\boldsymbol{\varphi}_\nu\}$ spannt *einen Vektorraum* V *auf, wenn sich jedes Element aus* V *als Linearkombination von Vektoren aus* $\widetilde{\varphi}$ *darstellen lässt. Die Koeffizienten der Linearkombination sind die* **Entwicklungskoeffizienten**.

Wenn zu jedem Element aus V eindeutige *Entwicklungskoeffizienten gehören, so bildet* $\widetilde{\varphi}$ *eine* **Basis** *für* V.

Die Menge von Vektoren ist orthogonal, *wenn gilt*

$$\boldsymbol{\varphi}_\mu^{\mathsf{T}}\boldsymbol{\varphi}_\nu = \langle \boldsymbol{\varphi}_\mu, \boldsymbol{\varphi}_\nu \rangle = \begin{cases} \alpha_{\mu,\nu} & : \quad \mu = \nu \ , \\ 0 & : \quad \textit{sonst} \ . \end{cases} \tag{3.2.1}$$

Wenn (3.2.1) gilt, heißt $\widetilde{\varphi}$ *eine* **orthogonale Basis**, *und wenn* $\alpha_{\mu,\nu} = 1$ *ist, eine* **orthonormale Basis**.

Die Entwicklung eines Vektors $\boldsymbol{f}$ nach *orthonormalen* Basisvektoren aus $\widetilde{\varphi}$ ergibt die *Entwicklungskoeffizienten*

$$\boxed{c_\nu = \boldsymbol{\varphi}_\nu^{\mathsf{T}}\boldsymbol{f} = \langle \boldsymbol{f}, \boldsymbol{\varphi}_\nu \rangle = \sum_{j=0}^{M-1} \varphi_{\nu j} f_j \quad \text{oder} \quad \boldsymbol{c} = \boldsymbol{\Phi}\boldsymbol{f} \ ,} \tag{3.2.2}$$

wenn man die orthonormalen Basisvektoren $\boldsymbol{\varphi}_\nu^{\mathsf{T}}$ den Zeilen der Matrix $\boldsymbol{\Phi}$ zuordnet. Bei dieser Vorgehensweise wird also für T_{r} in (3.1.1) eine *lineare Transformation* gewählt, wobei linear im Sinne von (2.3.2), S. 87, zu verstehen ist. Die Vektoren $\boldsymbol{f}$ und $\boldsymbol{c}$ haben M und n Komponenten, $M \geq n$, sodass die Matrix $\boldsymbol{\Phi}$ die Größe nM hat. Die Umkehrung von (3.2.2) bzw. die Rekonstruktion von $\boldsymbol{f}$ ergibt sich aus

$$\boxed{\widehat{\boldsymbol{f}} \ = \ \boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{c} = \sum_{\nu=1}^{n} c_\nu \boldsymbol{\varphi}_\nu \ .} \tag{3.2.3}$$

Dabei ist $\widehat{\boldsymbol{f}}$ eine Approximation für $\boldsymbol{f}$, und für $n = M$ ist $\widehat{\boldsymbol{f}} = \boldsymbol{f}$, wenn die Basisvektoren vollständig sind. Es gilt der folgende Satz.

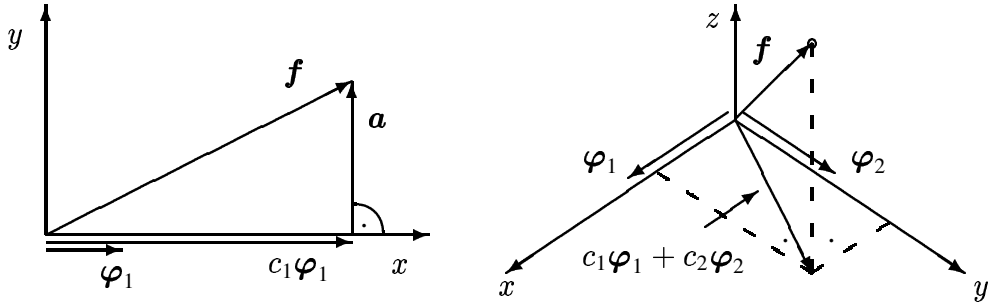


Bild 3.2.1: Zum Beweis von Satz 3.1 mit dem Orthogonalitätsprinzip

Satz 3.1 *Der mittlere quadratische Fehler*

$$\varepsilon = (\boldsymbol{f} - \widehat{\boldsymbol{f}})^{\mathsf{T}}(\boldsymbol{f} - \widehat{\boldsymbol{f}}) \tag{3.2.4}$$

der Approximation von $\boldsymbol{f}$ durch $\widehat{\boldsymbol{f}}$ gemäß (3.2.3) wird minimiert, wenn man den Vektor $\boldsymbol{c}$ gemäß (3.2.2) wählt. Für $n = M$ ist $\varepsilon = 0$, und es gilt

$$\boldsymbol{f} = \sum_{\nu=1}^{M} c_\nu \boldsymbol{\varphi}_\nu = \sum_{\nu=1}^{M} \langle \boldsymbol{f}, \boldsymbol{\varphi}_\nu \rangle \boldsymbol{\varphi}_\nu \; . \tag{3.2.5}$$

Beweis: Man findet einen Beweis dieses Satzes z. B. in [Albert, 1972]. Er lässt sich unmittelbar anschaulich führen, wenn man nur einen Vektor $\boldsymbol{\varphi}_1$ wie in Bild 3.2.1 betrachtet. Die Forderung nach Minimierung von (3.2.4) besagt dann, dass der Abstand zwischen $\boldsymbol{f}$ und $c_1\boldsymbol{\varphi}_1$ minimal sein soll. Das ist dann der Fall, wenn der Vektor

$$\boldsymbol{a} = \boldsymbol{f} - c_1\boldsymbol{\varphi}_1$$

senkrecht auf $\boldsymbol{\varphi}_1$ steht, also wenn

$$\boldsymbol{\varphi}_1^{\mathsf{T}}(\boldsymbol{f} - c_1\boldsymbol{\varphi}_1) = 0$$

ist. Verallgemeinert lässt sich sagen, dass mit $n < M$ Vektoren $\boldsymbol{\varphi}_\nu, \nu = 1, \ldots, n$ der Fehler dann am kleinsten ist, wenn $\boldsymbol{a}$ senkrecht auf dem durch die $\boldsymbol{\varphi}_\nu$ aufgespannten Raum steht, d. h. es gilt das **Orthogonalitätsprinzip**

$$\begin{aligned} \boldsymbol{\Phi}(\boldsymbol{f} - \boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{c}) &= 0 \; , \\ \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{c} &= \boldsymbol{\Phi}\boldsymbol{f} \; , \\ \boldsymbol{c} &= \boldsymbol{\Phi}\boldsymbol{f} \; . \end{aligned}$$

Die letzte Gleichung folgt aus der Orthonormalität der Basisvektoren, da dann $\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathsf{T}}$ die $n \times n$ Einheitsmatrix ergibt. Der letzte Teil von Satz 3.1 enthält nur die bekannte Aussage, dass sich ein Vektor $\boldsymbol{f} \in \mathbb{R}^M$ nach einer Orthonormalbasis entwickeln lässt. Damit ist Satz 3.1 bewiesen.

Die obigen Beziehungen wurden diskret, d. h. für Vektoren, formuliert. Analoge Beziehungen gelten für die Entwicklung von Funktionen $f(t)$ bzw. $f(\boldsymbol{x})$ nach einem orthonormalen

System von Basisfunktionen $\varphi_\nu(t)$. Insbesondere gilt in Analogie von (3.2.5)

$$\boxed{\begin{aligned} f(t) &= \sum_{\nu=-\infty}^{\infty} c_\nu \varphi_\nu(t) \\ &= \sum_{\nu} \langle f(\tau), \varphi_\nu(\tau)\rangle \varphi_\nu(t) \ , \\ c_\nu &= \int_{-\infty}^{\infty} f(\tau)\varphi_\nu(\tau)\,\mathrm{d}\tau = \langle f(\tau), \varphi_\nu(\tau)\rangle \ . \end{aligned}} \tag{3.2.6}$$

Dabei wird die quadratische Integrierbarkeit von $f(t)$ vorausgesetzt, d. h.

$$\int_{-\infty}^{\infty} |f(t)|^2\,\mathrm{d}t < \infty \ . \tag{3.2.7}$$

Der Vektorraum der eindimensionalen, quadratisch integrierbaren Funktionen wird mit $\mathsf{L}^2(\mathbb{R})$ bezeichnet, wobei $\mathbb{R}$ die rellen Zahlen sind.

Die Vektoren eines Basissystems sind meistens in bestimmter Weise geordnet, z. B. die $\boldsymbol{\varphi}_\nu$ in (3.2.15) nach steigenden Frequenzen. Natürlich bedeutet das i. Allg. *nicht*, dass die n ersten Entwicklungskoeffizienten in (3.2.2) deswegen die für die Klassifikation am besten – oder am schlechtesten – geeigneten Merkmale sind. Eine Merkmalsbewertung und –auswahl mit den in Abschnitt 3.9 beschriebenen Verfahren wird also trotzdem zweckmäßig sein.

Man kann eine Entwicklung entweder auf das *ganze Objekt* oder nur auf die *Konturlinie* dieses Objekts anwenden. Im ersten Falle haben die globalen Objekteigenschaften einen starken Einfluss, im zweiten werden kleinere Einzelheiten stärker bewichtet. Beide Vorgehensweisen finden Anwendung. Beispiele für Orthogonaltransformationen folgen in Abschnitt 3.2.2 bis 3.2.6; daneben sind aus der mathematischen Literatur verschiedene orthogonale Polynome bekannt.

In der Signalverarbeitung wird die Entwicklung (3.2.2) auch als *Analyseteil* und die Rekonstruktion (3.2.5) auch als *Syntheseteil* bezeichnet. Aus den Gleichungen geht hervor, dass die Aufeinanderfolge von Analyse und Synthese die *Identität* ergibt. Es handelt sich um *lineare Operationen*, die durch Vektor– und Matrixoperationen realisiert werden.

Eine Verallgemeinerung der Entwicklung nach orthogonalen Basisvektoren ist die Entwicklung nach einer biorthogonalen Basis.

Definition 3.2 *Zwei Mengen von Vektoren $\widetilde{\varphi} = \{\boldsymbol{\varphi}_\nu\}$ und $\widetilde{\chi} = \{\boldsymbol{\chi}_\nu\}$ heißen* **biorthogonale Basis** *(oder auch* **duale Basis***), wenn Definition 3.1 gilt mit der Verallgemeinerung*

$$\boldsymbol{\varphi}_\mu^{\mathsf{T}} \boldsymbol{\chi}_\nu = \langle \boldsymbol{\varphi}_\mu, \boldsymbol{\chi}_\nu \rangle = \begin{cases} \alpha_{\mu,\nu} & : \ \mu = \nu \ , \\ 0 & : \ \textit{sonst} \ . \end{cases} \tag{3.2.8}$$

Die Menge $\widetilde{\chi}$ wird auch die zu $\widetilde{\varphi}$ **duale Menge** von Vektoren genannt. Dafür gilt in Verallgemeinerung von (3.2.5)

$$\boldsymbol{f} = \sum_{\nu=1}^{M} \langle \boldsymbol{f}, \boldsymbol{\chi}_\nu \rangle \boldsymbol{\varphi}_\nu \ . \tag{3.2.9}$$

Die Rahmen sind eine weitere Verallgemeinerung, die dadurch möglich wird, dass, wie schon erwähnt, Orthogonalität zwar eine nützliche und hinreichende Bedingung für die Entwicklung ist, aber keine notwendige.

Definition 3.3 *Eine Menge $\widetilde{\psi} = \{\boldsymbol{\psi}_\nu\}$ von Vektoren bildet einen* **Rahmen** *("frame"), wenn mit zwei Konstanten $A, B > 0$ (den* RIESZ*-Schranken) für alle Vektoren $\boldsymbol{f}$ gilt*

$$A\|\boldsymbol{f}^\mathsf{T}\boldsymbol{f}\|^2 \leq \sum_\nu \|\boldsymbol{f}^\mathsf{T}\boldsymbol{\psi}_\nu\|^2 \leq B\|\boldsymbol{f}^\mathsf{T}\boldsymbol{f}\|^2 \; . \tag{3.2.10}$$

Für $A = B$ liegt ein **enger Rahmen** *("tight frame") vor.*

Ein Operator $\boldsymbol{\Psi}$ heißt **Rahmenoperator** *eines Rahmens $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n\}$, wenn für alle Vektoren $\boldsymbol{f}$ gilt*

$$\boldsymbol{\Psi}\boldsymbol{f} = \sum_{\nu=1}^{n} \langle \boldsymbol{f}, \boldsymbol{\psi}_\nu \rangle \boldsymbol{\psi}_\nu \; . \tag{3.2.11}$$

Für orhtonormale Basen gilt $A = B = 1$, d. h. sie sind spezielle Rahmen. Ohne Beweis sei erwähnt, dass sich für einen Vektor $\boldsymbol{f}$ die Rekonstruktion

$$\boldsymbol{f} = \sum_\nu \langle \boldsymbol{f}, \boldsymbol{\Psi}^{-1}\boldsymbol{\psi}_\nu \rangle \boldsymbol{\psi}_\nu \tag{3.2.12}$$

ergibt. Man kann also sowohl orthonormale Basen als auch Rahmen zur Repräsentation von Mustern (bzw. Signalen) verwenden. Zur Berechnung des Rahmenoperators wird auf die Literatur in Abschnitt 3.11 verwiesen.

Die diskrete Wavelet Transformation in Abschnitt 3.3 lässt sich als Rahmendarstellung auffassen. Allerdings wird dort der Schwerpunkt auf den Aspekt der Auflösungshierarchie gelegt werden.

3.2.2 Diskrete FOURIER-Transformation

Die **diskrete** FOURIER-**Transformation** (DFT), die in Satz 2.11, S. 93, eingeführt wurde, lässt sich als eine spezielle orthogonale Entwicklung auffassen. Betrachtet man der Einfachheit halber nur den eindimensionalen Fall, so ergibt sich aus (2.3.28) mit

$$W_M = \exp\left[\frac{-\mathrm{i}\,2\pi}{M}\right] \tag{3.2.13}$$

eine Periode $[F_\nu]$ der periodischen Folge $[\widetilde{F}_\nu]$ von Entwicklungskoeffizienten zu

$$F_\nu = \sum_{j=0}^{M-1} f_j W_M^{j\nu} \; , \quad \nu = 0, 1, \ldots, M-1 \; . \tag{3.2.14}$$

Mit dem Vektor

$$\boldsymbol{\varphi}_\nu = \left(W_M^0, W_M^\nu, W_M^{2\nu}, \ldots, W_M^{(M-1)\nu}\right)^\mathsf{T} \tag{3.2.15}$$

gilt also analog zu (3.2.2)

$$F_\nu = \boldsymbol{\varphi}_\nu^\mathsf{T}\boldsymbol{f} \; . \tag{3.2.16}$$

Die Vektoren $\boldsymbol{\varphi}_\nu$ mit komplexen Komponenten sind orthogonal, wenn man in diesem Fall (3.2.1) zu $\boldsymbol{\varphi}_\nu^\mathsf{T}\boldsymbol{\varphi}_\mu^*$ modifiziert und unter $\boldsymbol{\varphi}_\mu^*$ den zu $\boldsymbol{\varphi}_\mu$ konjugiert komplexen Vektor versteht.

Die Orthogonalität folgt dann unmittelbar aus (2.3.31), (2.3.32), jedoch ist $\boldsymbol{\varphi}_\nu^{\mathsf{T}}\boldsymbol{\varphi}_\nu^*$ nicht auf den Wert Eins normiert.

Betrachtet man statt der periodischen Folge $[\tilde{f}_j]$ die um m Abtastwerte verschobene Folge $[\tilde{f}'_j]$, deren Werte definiert sind durch

$$\tilde{f}'_j = \tilde{f}_{j+m} \; , \tag{3.2.17}$$

so gilt in Analogie zu (2.1.10), S. 64, der folgende Satz.

Satz 3.2 *Gegeben sei*

$$[\widetilde{F}_\nu] = \mathrm{DFT}\{[\tilde{f}_j]\} \; ; \tag{3.2.18}$$

dann gilt für die verschobene Folge in (3.2.17)

$$[\widetilde{F}'_\nu] = \mathrm{DFT}\{[\tilde{f}'_j]\} = [\widetilde{F}_\nu W_M^{-\nu m}] \; . \tag{3.2.19}$$

Beweis: Der Beweis erfolgt über die Definition der DFT. Für ein Element $\widetilde{F}'_\nu$ der Folge $[\widetilde{F}'_\nu]$ gilt

$$\widetilde{F}'_\nu \;=\; \sum_{j=0}^{M-1} \tilde{f}'_j W_M^{\nu j} = \sum_{j=0}^{M-1} \tilde{f}_{j+m} W_M^{\nu j} \; .$$

Die Substitution $j + m = l$ ergibt

$$\widetilde{F}'_\nu \;=\; \sum_{l=m}^{m+M-1} \tilde{f}_l W_M^{\nu(l-m)} = W_M^{-\nu m} \sum_{l} \tilde{f}_l W_M^{\nu l} \; . \tag{3.2.20}$$

Der wesentliche Punkt ist nun, dass $[\tilde{f}_l]$ eine periodische Folge und

$$W_M^{\nu j} = W_M^{\nu(j+kM)} \; , \quad k = 0, \pm 1, \pm 2, \ldots \tag{3.2.21}$$

ist. Daher gilt auch

$$\widetilde{F}'_\nu = W_M^{-\nu m} \sum_{l=0}^{M-1} \tilde{f}_l W_M^{\nu l} = W_M^{-\nu m} \mathrm{DFT}\{[\tilde{f}_\nu]\} \; . \tag{3.2.22}$$

Dieses geht auch anschaulich aus Bild 3.2.2 hervor. Damit ist Satz 3.2 bewiesen.

Ordnet man die Abtastwerte $[f_j]$ eines Musters mit (2.3.24) einer Periode von $[\tilde{f}_j]$ zu, so entspricht (3.2.17) einer Translation des Musters, die allerdings nur sinnvoll ist, wenn die Werte von $[f'_j]$ in der gleichen Periode von $[\tilde{f}_j]$ liegen wie die von $[f_j]$. Ein solcher Fall ist in Bild 3.2.2 unten gezeigt. Die Bedeutung eines Musters ist aber i. Allg. unabhängig von einer Translation. Daher werden als Merkmale c_ν oft nicht die Koeffizienten F_ν verwendet sondern

$$c_\nu = |F_\nu|^2 = F_\nu F_\nu^* \; . \tag{3.2.23}$$

Die Abtastwerte von Signalen, insbesondere von Bild– oder Sprachsignalen, sind ganze Zahlen, also *reelle Werte*; für diese gilt $|F_\nu|^2 = |F_{-\nu}|^2$. Bei einer Verschiebung hat man Merkmale

$$c'_\nu = |F'_\nu|^2 = |F_\nu W_M^{-\nu m}|^2 = |F_\nu|^2 = c_\nu \; , \tag{3.2.24}$$

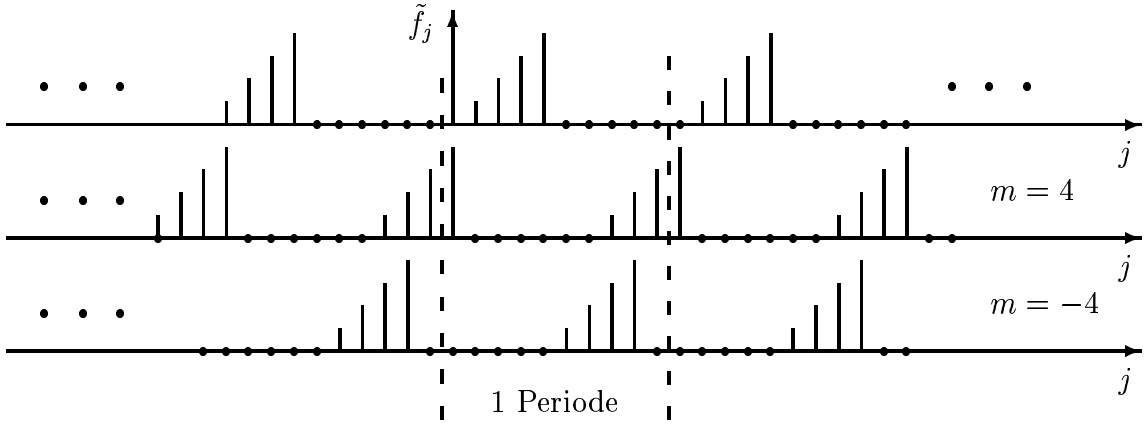


Bild 3.2.2: Verschiebung einer periodischen Folge

die eine **Translationsinvarianz** aufweisen. Mit (3.2.23) wird alle Phaseninformation eliminiert, nicht nur die durch die Translation verursachte zusätzliche Phasendrehung. Nun ist aber das Auge phasenempfindlich (anders als das Ohr). Das bedeutet, dass die in mit (3.2.23) gewonnenen Merkmalen enthaltene Information nicht ausreicht, ein optisches Muster zu rekonstruieren. Hier ist an die Bemerkungen von Abschnitt 3.1 zu erinnern. Danach kommt es darauf an, Merkmale zu finden, die eine Klassifikation ermöglichen, und das ist mit Merkmalen gemäß (3.2.23) erfahrungsgemäß in vielen Fällen möglich; die Rekonstruktion ist dagegen nicht von primärem Interesse.

Wenn sich ein Muster $\boldsymbol{f}$ gemäß

$$\boldsymbol{f} = \boldsymbol{s} * \boldsymbol{n} \tag{3.2.25}$$

aus der Faltung eines idealen Signals $\boldsymbol{s}$ mit einem Störanteil $\boldsymbol{n}$ zusammensetzt, werden Merkmale statt aus der DFT auch aus dem *Cepstrum* gewonnen. Für dieses sind verschiedene Definitionen in Gebrauch. Zunächst gilt für die additive Überlagerung

$$\boldsymbol{f} = \boldsymbol{s} + \boldsymbol{n} \tag{3.2.26}$$

von Signal und Störung, dass auch die Koeffizienten F_ν der DFT sich additiv überlagern

$$F_\nu = \mathrm{DFT}\{\boldsymbol{f}\} = \mathrm{DFT}\{\boldsymbol{s}\} + \mathrm{DFT}\{\boldsymbol{n}\} = S_\nu + N_\nu \ . \tag{3.2.27}$$

Dieses war die Grundlage für die Trennung von $\boldsymbol{s}$ und $\boldsymbol{n}$ mit einem linearen System in Abschnitt 2.3.4. Bei einer Überlagerung gemäß (3.2.25) wird man daher zunächst bestrebt sein, Signale $\boldsymbol{f}^+, \boldsymbol{s}^+, \boldsymbol{n}^+$ zu finden, die sich ebenfalls additiv überlagern. Dieses wird mit dem *komplexen Cepstrum*

$$\boldsymbol{f}^+ = \mathrm{DFT}^{-1}\{\log\left[\mathrm{DFT}\{\boldsymbol{f}\}\right]\} \tag{3.2.28}$$

erreicht. Wegen Satz 2.12, S. 96, gilt nämlich für $\boldsymbol{f}$ in (3.2.25)

$$\begin{aligned} \mathrm{DFT}\{\boldsymbol{f}\} &= \mathrm{DFT}\{\boldsymbol{s}\} \cdot \mathrm{DFT}\{\boldsymbol{n}\} \ , \\ \log\left[\mathrm{DFT}\{\boldsymbol{f}\}\right] &= \log\left[\mathrm{DFT}\{\boldsymbol{s}\}\right] + \log\left[\mathrm{DFT}\{\boldsymbol{n}\}\right] \ . \end{aligned}$$

Definiert man $\boldsymbol{s}^+$ und $\boldsymbol{n}^+$ analog zu $\boldsymbol{f}^+$ in (3.2.28), so gilt

$$\boldsymbol{f}^+ = \boldsymbol{s}^+ + \boldsymbol{n}^+ \ . \tag{3.2.29}$$

Grundsätzlich gelten auch hier die Bemerkungen von Abschnitt 2.3.4, wonach die DFT für *periodische Folgen* eingeführt wurde. Eine genauere Diskussion dieser Fragen im Zusammenhang mit dem komplexen Cepstrum erfolgt in der angegebenen Literatur. Als Anhaltspunkt kann dienen, dass die Verwendung endlicher Folgen in (3.2.28) mit genügender Genauigkeit zulässig ist, wenn M in (3.2.14) genügend groß ist, d. h. man muss gegebenenfalls eine endliche Folge $[f_j]$ durch Anhängen von Nullen verlängern. Es sei noch angemerkt, dass man die mit (3.2.25), (3.2.28), (3.2.29) angegebene Vorgehensweise auch zur Vorverarbeitung von Mustern anwendet. Die zugehörigen Systeme werden als *homomorphe Systeme* bezeichnet.

Da die DFT in (3.2.28) i. Allg. komplexe Koeffizienten ergibt, ist auch der komplexe Logarithmus zu verwenden. Um diesen zu vermeiden und aus den im Anschluss an (3.2.23) diskutierten Gründen wird daher statt des komplexen Cepstrums $\boldsymbol{f}^+$ auch das Cepstrum $\boldsymbol{f}^0$ verwendet, das definiert ist durch

$$\boldsymbol{f}^0 = \mathrm{DFT}^{-1}\{\log[|\mathrm{DFT}\{\boldsymbol{f}\}|^2]\} \; . \tag{3.2.30}$$

Ist F_ν ein Koeffizient der DFT von $[\boldsymbol{f}]$ gemäß (3.2.14), so wird also als Merkmal

$$c_\nu = \mathrm{DFT}^{-1}\left\{\log\left[|F_\nu|^2\right]\right\} \tag{3.2.31}$$

verwendet. Gebräuchlich sind auch

$$c_\nu = \log\left[|F_\nu|^2\right] \; , \tag{3.2.32}$$

$$c_\nu = \left|\log\left[|F_\nu|^2\right]\right|^2 \; . \tag{3.2.33}$$

Die Folge $[|F_\nu|^2]$ wird auch als **Leistungsspektrum** von $\boldsymbol{f}$ bezeichnet. Es ist die DFT der Autokorrelationsfunktion von $\boldsymbol{f}$ (s. (2.1.14), S. 64), sodass man letztere mit der DFT berechnen kann, wenn ähnliche Bedingungen, wie sie im Zusammenhang mit (2.3.37) diskutiert wurden, beachtet werden.

Es wurde bereits im Abschnitt 2.3.3 erwähnt, dass es schnelle Algorithmen zur Berechnung der DFT gibt. Zwar würde deren ausführliche Erörterung hier zu weit führen, jedoch lässt sich das Prinzip kurz darstellen. Fasst man die Koeffizienten F_ν der DFT in (3.2.14), (3.2.16) im Vektor $\boldsymbol{F}$ zusammen, so gilt

$$\boldsymbol{F} = \boldsymbol{W}_M \boldsymbol{f} \; , \tag{3.2.34}$$

$$\boldsymbol{W}_M = \left(W_M^{\nu j}\right) \; , \quad \nu, j = 0, 1, \ldots, M-1 \; . \tag{3.2.35}$$

Wegen (3.2.21) lassen sich alle Elemente von $\boldsymbol{W}_M$ so modulo M reduzieren, dass nur noch Elemente $W_M^k, \quad 0 \le k \le M-1$ auftreten. Diese Matrix wird $\widetilde{\boldsymbol{W}}_M$ genannt. Wir setzen nun voraus, dass $M = 2^q, \; q = 1, 2, \ldots$ ist. Die Zeilen von $\widetilde{\boldsymbol{W}}_M$ werden nach der Methode des "bit reversal" umgeordnet und ergeben eine Matrix $\widetilde{\boldsymbol{W}}'_M$. Ist beispielsweise $M = 8$, so werden die acht Zeilen von $\widetilde{\boldsymbol{W}}_M$ dezimal von 0 bis 7, binär von 000 bis 111 durchnummeriert. Die Zeile 3 mit der binären Darstellung 011 ergibt nach dem "bit reversal" binär 110 oder dezimal 6. Also wird Zeile 3 von $\widetilde{\boldsymbol{W}}_M$ die Zeile 6 von $\widetilde{\boldsymbol{W}}'_M$. Das Prinzip der **schnellen** FOURIER-**Transformation** (FFT, "Fast FOURIER-Transform") beruht darauf, dass sich $\widetilde{\boldsymbol{W}}'_M$ wie im folgenden Satz angegeben *faktorisieren* lässt.

Satz 3.3 *Die Matrix $\widetilde{\boldsymbol{W}}'_M$, die man durch Umordnung der Zeilen von $\widetilde{\boldsymbol{W}}_M$ nach der Methode des "bit reversal" erhält, lässt sich faktorisieren in*

$$\widetilde{\boldsymbol{W}}'_M = \begin{pmatrix} \widetilde{\boldsymbol{W}}'_{\frac{M}{2}} & \boldsymbol{O}_{\frac{M}{2}} \\ \boldsymbol{O}_{\frac{M}{2}} & \widetilde{\boldsymbol{W}}'_{\frac{M}{2}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{\frac{M}{2}} & \boldsymbol{O}_{\frac{M}{2}} \\ \boldsymbol{O}_{\frac{M}{2}} & \boldsymbol{K}_{\frac{M}{2}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{\frac{M}{2}} & \boldsymbol{I}_{\frac{M}{2}} \\ \boldsymbol{I}_{\frac{M}{2}} & -\boldsymbol{I}_{\frac{M}{2}} \end{pmatrix} . \tag{3.2.36}$$

Beweis*: s. z. B. [Pease, 1968]*

Dabei sind $\boldsymbol{O}_{\frac{M}{2}}$ und $\boldsymbol{I}_{\frac{M}{2}}$ Null– und Einheitsmatrizen der Größe $\frac{M}{2} \times \frac{M}{2}$. Die Matrix $\widetilde{\boldsymbol{W}}'_{\frac{M}{2}}$ erhält man aus

$$\boldsymbol{W}_{\frac{M}{2}} = \left(W_{\frac{M}{2}}^{\nu j}\right) = \left(W_M^{2\nu j}\right) \;; \quad \nu, j = 0, 1, \ldots, \frac{M}{2} - 1 \tag{3.2.37}$$

nach dem oben für $\boldsymbol{W}_M$, $\widetilde{\boldsymbol{W}}_M$ und $\widetilde{\boldsymbol{W}}'_M$ beschriebenen Verfahren. Die Diagonalmatrix $\boldsymbol{K}_{\frac{M}{2}}$ ist definiert durch

$$\boldsymbol{K}_{\frac{M}{2}} = \operatorname{diag}\left(W_M^{\nu}\right) \;, \quad \nu = 0, 1, \ldots, \frac{M}{2} - 1 \;. \tag{3.2.38}$$

Nach einmaliger Anwendung von (3.2.36) wird diese Gleichung erneut auf $\widetilde{\boldsymbol{W}}'_{\frac{M}{2}}$ angewendet. Das ergibt

$$\widetilde{\boldsymbol{W}}'_{\frac{M}{2}} = \begin{pmatrix} \widetilde{\boldsymbol{W}}'_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{4}} \\ \boldsymbol{O}_{\frac{M}{4}} & \widetilde{\boldsymbol{W}}'_{\frac{M}{4}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{4}} \\ \boldsymbol{O}_{\frac{M}{4}} & \boldsymbol{K}_{\frac{M}{4}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{\frac{M}{4}} & \boldsymbol{I}_{\frac{M}{4}} \\ \boldsymbol{I}_{\frac{M}{4}} & -\boldsymbol{I}_{\frac{M}{4}} \end{pmatrix} , \tag{3.2.39}$$

wobei zu beachten ist, dass gemäß (3.2.37)

$$\boldsymbol{K}_{\frac{M}{4}} = \operatorname{diag}\left(W_{\frac{M}{2}}^{\nu}\right) \;, \quad \nu = 0, 1, \ldots, \frac{M}{4} - 1 \tag{3.2.40}$$

ist. Setzt man (3.2.39) in (3.2.36) ein, so erhält man

$$\begin{aligned} \widetilde{\boldsymbol{W}}'_M \;=\; & \begin{pmatrix} \widetilde{\boldsymbol{W}}'_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{4}} & & \\ \boldsymbol{O}_{\frac{M}{4}} & \widetilde{\boldsymbol{W}}'_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{2}} & \\ & & \widetilde{\boldsymbol{W}}'_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{4}} \\ \boldsymbol{O}_{\frac{M}{2}} & & \boldsymbol{O}_{\frac{M}{4}} & \widetilde{\boldsymbol{W}}'_{\frac{M}{4}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{4}} & & \\ \boldsymbol{O}_{\frac{M}{4}} & \boldsymbol{K}_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{2}} & \\ & & \boldsymbol{I}_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{4}} \\ \boldsymbol{O}_{\frac{M}{2}} & & \boldsymbol{O}_{\frac{M}{4}} & \boldsymbol{K}_{\frac{M}{4}} \end{pmatrix} \\ & \begin{pmatrix} \boldsymbol{I}_{\frac{M}{4}} & \boldsymbol{I}_{\frac{M}{4}} & & \\ \boldsymbol{I}_{\frac{M}{4}} & -\boldsymbol{I}_{\frac{M}{4}} & \boldsymbol{O}_{\frac{M}{2}} & \\ & & \boldsymbol{I}_{\frac{M}{4}} & \boldsymbol{I}_{\frac{M}{4}} \\ \boldsymbol{O}_{\frac{M}{2}} & & \boldsymbol{I}_{\frac{M}{4}} & -\boldsymbol{I}_{\frac{M}{4}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{\frac{M}{2}} & \boldsymbol{O}_{\frac{M}{2}} \\ \boldsymbol{O}_{\frac{M}{2}} & \boldsymbol{K}_{\frac{M}{2}} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{\frac{M}{2}} & \boldsymbol{I}_{\frac{M}{2}} \\ \boldsymbol{I}_{\frac{M}{2}} & -\boldsymbol{I}_{\frac{M}{2}} \end{pmatrix} . \end{aligned} \tag{3.2.41}$$

Dieser Prozess wird fortgesetzt, bis er nach $(q-1)$ Schritten mit einer Matrix

$$\widetilde{\boldsymbol{W}}'_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \widetilde{\boldsymbol{W}}'_{\frac{M}{M/2}} = \left(W_M^{(M\nu j)/2}\right) \;, \quad \nu, j = 0, 1 \tag{3.2.42}$$

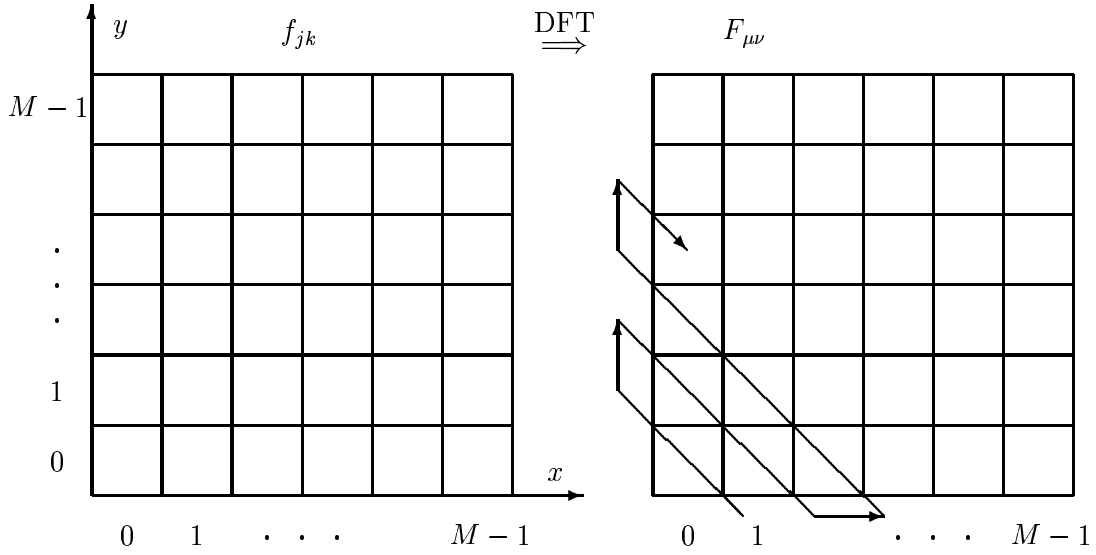


Bild 3.2.3: Zur Auswahl von FOURIER-Koeffizienten einer zweidimensionalen Folge von Abtastwerten

endet. Eine Berechnung von $\boldsymbol{F}$ nach (3.2.34) erfordert $\mathcal{O}(M^2)$ komplexe Multiplikationen. Bei einer vollständigen Faktorisierung von M mit (3.2.36) treten komplexe Multiplikationen nur noch in den Diagonalmatrizen $\boldsymbol{K}_{\frac{M}{2}}, \boldsymbol{K}_{\frac{M}{4}}, \ldots, \boldsymbol{K}_2$ auf. Sonst sind nur noch Additionen und Subtraktionen erforderlich. Die Zahl der komplexen Multiplikationen wird also auf $\mathcal{O}(M(\mathrm{ld}[M]-1)) \simeq \mathcal{O}(M\,\mathrm{ld}[M])$ reduziert. Im übrigen wird auf die in Abschnitt 3.11 zitierte Literatur verwiesen. Mit (3.2.36) hat man daher eine sehr effiziente Möglichkeit, Merkmale wie in (3.2.23), (3.2.31) zu berechnen. Man kann auf der Basis der FOURIER-Transformation nicht nur translationsinvariante, sondern auch skalen– (größen–) und rotationsinvariante Merkmale gewinnen. Die **Skaleninvarianz** erreicht man über eine MELLIN-Transformation, die **Rotationsinvarianz** durch Übergang auf Polarkoordinaten, da in diesen eine *Rotation* des Objekts und damit des Spektrums einer *Translation* im Winkel entspricht. Die Normierungsmaßnahmen von Abschnitt 2.5 bieten dazu eine Alternative.

Die DFT einer Folge mit M Abtastwerten liefert M FOURIER-Koeffizienten, von denen nur $n < M$ verwendet werden. Die Zahl n wird experimentell in Abhängigkeit von der Fehlerrate bei der Klassifikation festgelegt. Bei einer eindimensionalen Folge verwendet man entweder einfach die ersten n Koeffizienten und lässt die anderen weg oder bestimmt sie durch eine Merkmalsauswahl nach Abschnitt 3.9. Bei einer zweidimensionalen Folge sollten die Frequenzanteile in beiden Dimensionen in etwa zu gleichen Anteilen berücksichtigt werden. Daher werden die ersten n Koeffizienten z. B. durch eine diagonale Auswahl wie in Bild 3.2.3 bestimmt. Schließlich nimmt man insbesondere bei zweidimensionalen Folgen $[f_{jk}]$ vielfach nicht die Koeffizienten $[F_{\mu\nu}]$ der DFT oder deren Beträge selbst (3.2.23), sondern bildet die Summe über solche Werte (μ, ν), welche angenähert keil–, ring– oder balkenförmige Bereiche im rechtwinkligen Gitter der Koeffizienten $[F_{\mu\nu}]$ ergeben.

In Abschnitt 3.2.1 wurde bereits erwähnt, dass man eine Reihenentwicklung entweder auf das Objekt oder seine (geschlossene) Konturlinie anwenden kann. Für die DFT der Konturlinie lassen sich die Punkte (x_j, y_k) auf der Kontur als Muster $\boldsymbol{f}$ auffassen und in (3.2.14) verwenden, jedoch wurden dafür auch zwei andere Verfahren entwickelt. Danach fasst man entweder die Kontur als Funktion $u(t)$ in der komplexen Ebene auf oder stellt die Winkeländerung der

Tangente an die Kontur als Funktion $\alpha(l)$ der Bogenlänge l dar, wie es in Bild 3.2.4a,b angedeutet ist. Im ersteren Falle entwickelt man $u(t)$ in eine FOURIER-Reihe, deren Koeffizienten a_ν allerdings vom Startpunkt, sowie von Translation, Rotation und Skalierung abhängen. Dagegen sind die Koeffizienten

$$b_{\mu\nu} = \frac{a_{1+\mu}^{\nu/\kappa} a_{1-\nu}^{\mu/\kappa}}{a_1^{(\mu+\nu)/\kappa}} \tag{3.2.43}$$

von diesen Transformationen unabhängig, wenn κ der gemeinsame Faktor von μ und ν ist. Im letzteren Falle geht man von $\alpha(l)$ zunächst auf eine normierte Funktion

$$\alpha^*(t) = \alpha\frac{tL}{2\pi} + t\ , \quad 0 \le t \le 2\pi \tag{3.2.44}$$

über, wobei L die Bogenlänge der geschlossenen Konturlinie ist. Damit sind alle ebenen, einfach geschlossenen Kurven mit Startpunkt in die Klasse der in $(0, 2\pi)$ periodischen, gegen Translation, Rotation und Skalierung der Kontur invarianten Funktionen abgebildet. Die Entwicklung von α^* in eine FOURIER-Reihe ergibt

$$\begin{aligned} \alpha^*(t) &= \alpha_0 + \sum_{n=1}^{\infty}(a_n \cos[nt] + b_n \sin[nt]) \\ &= \alpha_0 + \sum_{n=1}^{\infty} A_n \cos[nt - \beta_n]\ . \end{aligned} \tag{3.2.45}$$

Wenn man die Kontur mit einem Polygon approximiert, z. B. mit dem in Abschnitt 3.10 beschriebenen Verfahren, so erhält man mit den Bezeichnungen von Bild 3.2.4c

$$\begin{aligned} \alpha_0 &= -\pi - \frac{1}{L}\sum_{j=1}^{m} l_j \Delta\alpha_j\ , \quad \text{mit}\ \ l_j = \sum_{i=l}^{j} \Delta l_i\ , \\ a_n &= -\frac{1}{n\pi}\sum_{j=1}^{m} \Delta\alpha_j \sin\left[\frac{2\pi n l_j}{L}\right]\ , \\ b_n &= -\frac{1}{n\pi}\sum_{j=1}^{m} \Delta\alpha_j \cos\left[\frac{2\pi n l_j}{L}\right]\ . \end{aligned} \tag{3.2.46}$$

Man kann zeigen, dass man mit α_0, a_n, b_n oder mit den als FOURIER-Deskriptoren bezeichneten Größen α_0, A_n, β_n die Kontur rekonstruieren kann, wenn noch L, α_0, P_0 bekannt sind.

3.2.3 Gefensterte FOURIER-Transformation

Die DFT, bzw. ihr kontinuierliches Analogon, die FOURIER-Transformation (2.3.20), S. 92, wird auf den gesamten Definitionsbereich eines Musters angewendet. Das bedeutet, dass man eine Menge *globaler Merkmale* berechnet. Wenn sich ein kleines Detail im Muster ändert, ändern sich alle Merkmale, gleichgültig ob man z. B. ein Sprachsignal von 10msec oder 10sec Dauer oder ein Bild der Größe 64×64 oder 4096×4096 Bildpunkte hat. Um dieses zu vermeiden, wird oft auch die **gefensterte FOURIER-Transformation** verwendet. Dabei wird eine *Fensterfunktion* $w(x, y)$, die außerhalb eines Intervalls $0 \le x, y \le x_\mathrm{m}, y_\mathrm{m}$ identisch verschwindet, bzw. w_{jk}, die für Indizes $j, k < 0$ und $j, k \ge N_x, N_y$ Werte identisch Null hat, verwendet. Sie hat

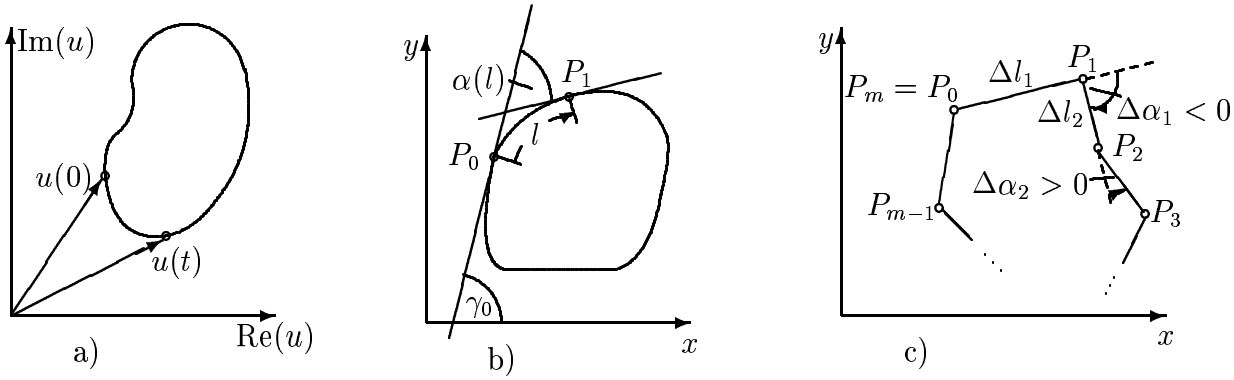


Bild 3.2.4: Zur DFT der Konturlinie

also eine *endliche* und in der Regel gegenüber der Mustergröße eine sehr *kleine* Ausdehnung. Der Anfang der Fensterfunktion wird auf einen Punkt (x_0, y_0) bzw. (j_0, k_0) des Musters geschoben. Die korrespondierenden Werte werden multipliziert und das Ergebnis, also ein kleiner Ausschnitt des Musters, transformiert. Fensterfunktionen wurden in (2.5.43), S. 131, angegeben, in (3.4.6), S. 198, wird die GAUSS-Funktion als Fensterfunktion verwendet. Die Wiederholung der Transformation an unterschiedlichen Stellen des Musters ergibt dann FOURIER-Koeffizienten, die *ortsabhängig* sind und nur Information über die FOURIER-Transformierte in einer *lokalen Nachbarschaft* des Musters enthalten. Man erhält damit eine Menge *lokaler Merkmale*. Die mit einem Fenster an der Stelle (x_0, y_0) bzw. j_0, k_0 berechnete FOURIER-Transformation bzw. DFT ist

$$F_{[x_0,y_0]}(\xi,\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)w(x-x_0,y-y_0)\exp[-\mathrm{i}\,2\pi(\xi x+\eta y)]\;\mathrm{d}x\,\mathrm{d}y \quad (3.2.47)$$

$$F_{[j_0,k_0];\,\mu,\nu} = \sum_{j=0}^{M_x-1}\sum_{k=0}^{M_y-1} f_{jk}\,w_{j-j_0,k-k_0}\exp\left[-\mathrm{i}\,2\pi\left(\frac{\mu j}{M_x}+\frac{\nu k}{M_y}\right)\right]. \quad (3.2.48)$$

Eine Modifikation der obigen Gleichungen besteht darin, sie nicht über den Definitionsbereich des Musters, sondern über den der Fensterfunktion, zu definieren.

Diese *gefensterten* oder, wenn es sich um Zeitfunktionen handelt, *Kurzzeittransformationen* spielen in der Bild– und Sprachverarbeitung eine wichtige Rolle, wie auch aus Abschnitt 3.4.1 und Abschnitt 3.6.1 hervorgeht. Natürlich ist das Prinzip der Fensterung nicht auf die DFT beschränkt, sondern kann auch bei anderen Transformationen analog genutzt werden. Ein weiteres Beispiel ist die GAUSS-gefensterte FOURIER-Transformation in (3.4.6), S. 198. Die Wahl der Fenstergröße ist anwendungsabhängig; das Fenster muss zum einen so klein sein, dass interessierende Änderungen im Muster nicht verschliffen werden, es muss zum anderen so groß sein, dass genügend lokale Information vorhanden ist.

3.2.4 Diskrete Cosinus Transformation

Im letzten Abschnitt wurde die Nutzung der diskreten FOURIER-Transformation für die Merkmalsberechnung diskutiert. Aus Abschnitt 2.3.3 geht hervor, dass sie für eine *periodische* diskrete Folge von Abtastwerten eingeführt wurde. Aus Satz 2.12, S. 96 geht ihre besondere Bedeutung für die Behandlung linearer Systeme hervor. Für die *Approximation* von Funktionen

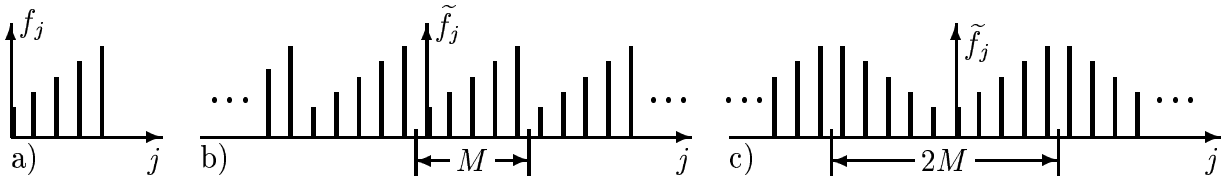


Bild 3.2.5: a) Eine Folge von Abtastwerten; b) deren periodische Fortsetzung bei der DFT, c) bzw. bei der DCT

durch eine orthogonale Reihenentwicklung und damit für die Merkmalsgewinnung hat dagegen die **diskrete cosinus Transformation** (DCT) deutliche Vorteile. Sie findet z. B. Einsatz im Rahmen der digitalen Bildübertragung und Kodierung. Bei ihrer Definition beschränken wir uns hier auf eine eindimensionale periodische Folge $[\widetilde{f}_j]$; natürlich wird man sich in Rechnungen, wie üblich, auf die Betrachtung nur *einer* Periode beschränken. Die DCT und ihre Umkehrung ergeben sich aus dem folgenden Satz.

Satz 3.4 *Definiert man die DCT einer (periodischen) Folge* $[\widetilde{f}_j]$ *mit*

$$\begin{aligned} \widetilde{F}_0 &= \frac{\sqrt{2}}{M}\sum_{j=0}^{M-1}\widetilde{f}_j\ , \\ \widetilde{F}_\mu &= \frac{2}{M}\sum_{j=0}^{M-1}\widetilde{f}_j\cos\left[\frac{(2j+1)\mu\pi}{2M}\right]\ , \quad \mu=1,2,\ldots,M-1\ , \end{aligned} \tag{3.2.49}$$

so erhält man die Folge $[\widetilde{f}_j]$ *aus der inversen Beziehung*

$$\widetilde{f}_j = \frac{1}{\sqrt{2}}\widetilde{F}_0 + \sum_{\mu=1}^{M-1}\widetilde{F}_\mu\cos\left[\frac{(2j+1)\mu\pi}{2M}\right]\ , \quad j=0,1,\ldots,M-1\ . \tag{3.2.50}$$

Vergleicht man (3.2.49) mit (2.3.28), S. 93, und berücksichtigt (3.2.51), so sieht man, dass die DCT gerade der Realteil einer DFT *doppelter* Periodenlänge ist. Da zudem $\cos\alpha$ eine gerade Funktion ist, sind auch die betrachteten periodischen Folgen $[\widetilde{f}_j]$, $[\widetilde{F}_\mu]$ gerade Folgen. Wie Bild 3.2.5 illustriert, wird also eine Folge $[f_j]$, $j=0,1,\ldots,M-1$ bei der DFT durch Wiederholen dieser Folge mit der Periodenlänge M periodisch fortgesetzt. Das führt i. Allg. zu Unstetigkeiten, die eine große Zahl von Koeffizienten zu ihrer Approximation erfordern. Im Unterschied dazu wird eine Folge $[f_j]$ bei der DCT zunächst an der y–Achse gespiegelt und dann diese nunmehr gerade Folge periodisch fortgesetzt, und zwar natürlich mit Periodenlänge $2M$. Wie man sieht, entfallen damit die Unstetigkeiten. Daraus resultieren die vorteilhaften Eigenschaften der DCT für die Approximation von Funktionen.

Die DCT ist eine Orthogonaltransformation und daher *abstandserhaltend*. Wie für die DFT gibt es auch für die DCT schnelle Algorithmen zu ihrer Berechnung. Eine eminent wichtige Eigenschaft ist, dass sie eine gute *Approximation der* KARHUNEN–LOÈVE*-Transformation* (KLT, s. Abschnitt 3.8.2) für stark korrelierte Daten ist. Ähnlich wie die KLT konzentriert die DCT die Energie eines Signals zum Großteil in den Koeffizienten niederer Ordnung. Diese Eigenschaft ist der Grund, warum die DCT eine starke Verbreitung im Rahmen der Datenreduktion gefunden hat und weshalb sie auch ein guter Kandidat für die Merkmalsgewinnung

ist.

3.2.5 WALSH-Transformation

Die Basisvektoren $\boldsymbol{\varphi}_\nu$ der DFT in (3.2.15) lassen sich wegen der bekannten Beziehung

$$\exp[\mathrm{i}\,\alpha] = \cos\alpha + \mathrm{i}\,\sin\alpha \tag{3.2.51}$$

in einen Real- und einen Imaginärteil mit der geraden Funktion $\cos\alpha$ und der ungeraden Funktion $\sin\alpha$ zerlegen. Eine ähnliche Entwicklung erlauben die WALSH-Funktionen, die aber wegen ihrer auf ± 1 beschränkten Werte weniger Rechenaufwand erfordern.

Definition 3.4 *Im kontinuierlichen Falle sind die* WALSH-**Funktionen** *rekursiv definiert durch*

$$\begin{aligned} \mathrm{wal}[x;2j+p] &= (-1)^{[j/2]+p}\left(\mathrm{wal}\left[2\left(x+\frac{1}{4}\right);j\right] + (-1)^{j+p}\mathrm{wal}\left[2\left(x-\frac{1}{4}\right);j\right]\right)\,, \\ \mathrm{wal}[x;0] &= \begin{cases} 1 & : \quad -\frac{1}{2} \le x \le \frac{1}{2} \\ 0 & : \quad \text{sonst}\,. \end{cases} \end{aligned} \tag{3.2.52}$$

In (3.2.52) ist $j = 0, 1, 2, \ldots$, $p = 0, 1$ und $[j/2]$ die größte ganze Zahl, die nicht größer als $j/2$ ist. Die Funktionen sind auf das Intervall $-\frac{1}{2} \le x \le \frac{1}{2}$ beschränkt. Setzt man

$$\varphi_\nu(x) = \mathrm{wal}\left[\frac{x}{x_0};\nu\right]\,, \tag{3.2.53}$$

so sind die $\varphi_\nu(x)$ auf das Intervall $-\frac{x_0}{2} \le x \le \frac{x_0}{2}$ beschränkt. Einige Funktionen sind in Bild 3.2.6 dargestellt. Ohne Beweis wird angemerkt, dass die WALSH-Funktionen orthonormal sind, d. h. es gilt

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} \mathrm{wal}[x;j]\,\mathrm{wal}[x;k]\,\mathrm{d}x = \begin{cases} 1 & : \quad j = k \\ 0 & : \quad \text{sonst} \end{cases}\,. \tag{3.2.54}$$

Für die digitale Verarbeitung wurden verschiedene diskrete Versionen vorgeschlagen, die sich teilweise nur in der Reihenfolge der Funktionen unterscheiden. Hier wird lediglich die sogenannte HADAMARD geordnete WALSH–HADAMARD-Transformation (WHT) erläutert. Die Vektoren φ_ν erhält man aus den WALSH-Funktionen von Bild 3.2.6, indem man das Intervall $(-\frac{1}{2}, \frac{1}{2})$ mit $M = 2^q$ Werten abtastet, wobei es nur Abtastwerte ± 1 gibt. Die Transformationsmatrix der Größe M^2 lässt sich rekursiv aus der HADAMARD-**Matrix**

$$\boldsymbol{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.2.55}$$

berechnen, die übrigens gleich $\widetilde{\boldsymbol{W}}_2'$ in (3.2.42) ist. Es gilt

$$\begin{aligned} \boldsymbol{H}_M &= \boldsymbol{H}_2 \otimes \boldsymbol{H}_{\frac{M}{2}}\,, \quad M = 2^q \\ &= \underbrace{\boldsymbol{H}_2 \otimes \boldsymbol{H}_2 \otimes \ldots \boldsymbol{H}_2}_{q \text{ Faktoren}}\,. \end{aligned} \tag{3.2.56}$$

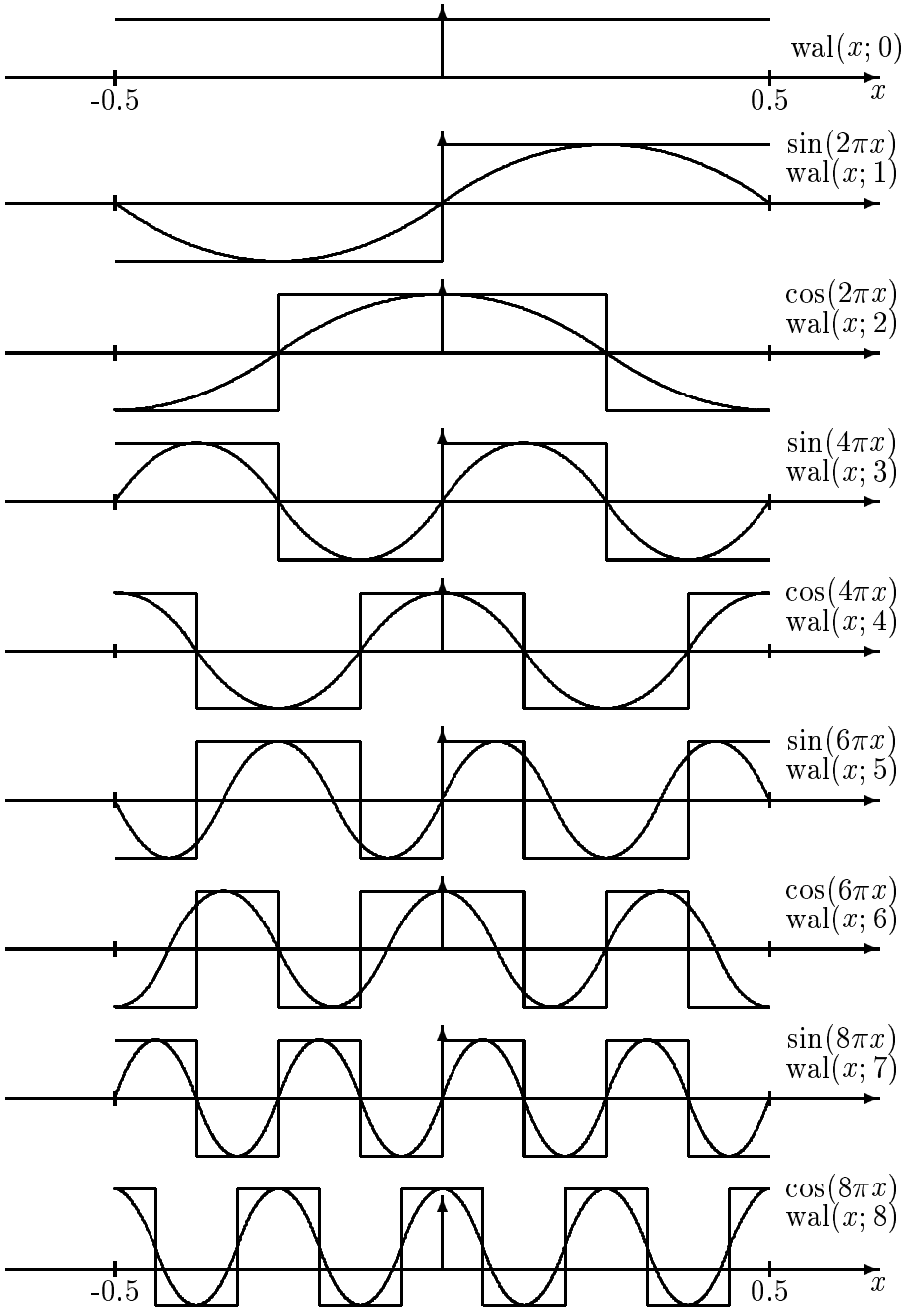


Bild 3.2.6: Einige WALSH-Funktionen und zum Vergleich entsprechende harmonische Funktionen

Dabei kennzeichnet $\otimes$ das KRONECKER-**Produkt** zweier Matrizen. Für die M^2 Matrix $\boldsymbol{A}$ und die m^2 Matrix $\boldsymbol{B}$ ist das Ergebnis eine $(Mm)^2$ Matrix

$$\boldsymbol{D} = \boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1M}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots & a_{2M}\boldsymbol{B} \\ \vdots & & & \\ a_{M1}\boldsymbol{B} & a_{M2}\boldsymbol{B} & \dots & a_{MM}\boldsymbol{B} \end{pmatrix} . \tag{3.2.57}$$

Zum Beispiel ist die HADAMARD-Matrix

$$\begin{aligned}\boldsymbol{H}_8 &= \boldsymbol{H}_2 \otimes \boldsymbol{H}_2 \otimes \boldsymbol{H}_2 = (\boldsymbol{H}_2 \otimes \boldsymbol{H}_2) \otimes \boldsymbol{H}_2 = \boldsymbol{H}_2 \otimes (\boldsymbol{H}_2 \otimes \boldsymbol{H}_2) \\ &= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \otimes \boldsymbol{H}_2 \\ &= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix} . \end{aligned} \tag{3.2.58}$$

Sie enthält die Abtastwerte der ersten acht WALSH-Funktionen, aber wie erwähnt in anderer Anordnung. Zudem ist bei einigen das Vorzeichen umgekehrt.

Definition 3.5 *Die* HADAMARD- *geordnete* WALSH–HADAMARD-**Transformation** (WHT) *eines Mustervektors* $\boldsymbol{f}$ *mit* M *Komponenten erfolgt gemäß*

$$\boldsymbol{c} = \boldsymbol{H}_M \boldsymbol{f} = \mathrm{WHT}\{\boldsymbol{f}\} \tag{3.2.59}$$

Die inverse WALSH–HADAMARD-*Transformation lautet*

$$\boldsymbol{f} = \frac{1}{M} \boldsymbol{H}_M \boldsymbol{c} = \mathrm{WHT}^{-1}\{\boldsymbol{c}\} \; . \tag{3.2.60}$$

Die WHT erfordert nur Additionen und Subtraktionen. Es gibt einen Algorithmus für die **schnelle WHT**, der sich ähnlich wie in (3.2.36) durch Faktorisierung der Transformationsmatrix $\boldsymbol{H}_M$ gewinnen lässt. Zur Abwechslung wird die Faktorisierung hier anschaulich über den **Signalflussgraph** angegeben. Als Beispiel betrachten wir die WHT für $M = 8$, für die man mit (3.2.58), (3.2.59) die Beziehung (3.2.61) erhält. Zur Unterscheidung der Zwischenergebnisse werden die Abtastwerte f_j mit einem weiteren Index l als f_j^l geschrieben. Mit $l = 0$ werden die Anfangswerte gekennzeichnet, also $f_j^0 = f_j,\ j = 0, 1, ..., M-1$, und mit $l = 1, 2, ..., q$ die Ergebnisse nach l Iterationen, wobei $f_j^q = c_j,\ j = 0, 1, ..., M-1$ das Endergebnis ist. Zerlegt man $\boldsymbol{c}$ und $\boldsymbol{f}$ entlang der gepunkteten Linie in (3.2.61), so gilt für die obere Hälfte die Beziehung (3.2.62).

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ . \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \begin{pmatrix} \boldsymbol{H}_4 & \boldsymbol{H}_4 \\ \cdots\cdots & \cdots\cdots \\ \boldsymbol{H}_4 & -\boldsymbol{H}_4 \end{pmatrix} \begin{pmatrix} f_0^0 \\ f_1^0 \\ f_2^0 \\ f_3^0 \\ .. \\ f_4^0 \\ f_5^0 \\ f_6^0 \\ f_7^0 \end{pmatrix} , \tag{3.2.61}$$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \boldsymbol{H}_4 \begin{pmatrix} f_0^0 \\ f_1^0 \\ f_2^0 \\ f_3^0 \end{pmatrix} + \boldsymbol{H}_4 \begin{pmatrix} f_4^0 \\ f_5^0 \\ f_6^0 \\ f_7^0 \end{pmatrix} = \boldsymbol{H}_4 \begin{pmatrix} f_0^1 \\ f_1^1 \\ f_2^1 \\ f_3^1 \end{pmatrix} , \tag{3.2.62}$$

$$f_j^1 = f_j^0 + f_{j+4}^0 \quad , \quad j = 0,1,2,3 \, .$$

Entsprechend erhält man für die untere Hälfte

$$\begin{pmatrix} c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \boldsymbol{H}_4 \begin{pmatrix} f_0^0 \\ f_1^0 \\ f_2^0 \\ f_3^0 \end{pmatrix} - \boldsymbol{H}_4 \begin{pmatrix} f_4^0 \\ f_5^0 \\ f_6^0 \\ f_7^0 \end{pmatrix} = \boldsymbol{H}_4 \begin{pmatrix} f_4^1 \\ f_5^1 \\ f_6^1 \\ f_7^1 \end{pmatrix} \tag{3.2.63}$$

$$f_j^1 = f_{j-4}^0 - f_j^0 \quad , \quad j = 4,5,6,7 \, ,$$

Nun hat aber wegen (3.2.58) $\boldsymbol{H}_4$ die Form

$$\boldsymbol{H}_4 = \boldsymbol{H}_2 \otimes \boldsymbol{H}_2 = \begin{pmatrix} \boldsymbol{H}_2 & \boldsymbol{H}_2 \\ \boldsymbol{H}_2 & -\boldsymbol{H}_2 \end{pmatrix} , \tag{3.2.64}$$

sodass sich die auf (3.2.61) angewendete Zerlegung auch auf (3.2.62), (3.2.63) anwenden lässt. Diese Zerlegung ist für $M = 8$ im Signalflussgraphen von Bild 3.2.7 dargestellt. Sie endet mit

$$\begin{pmatrix} c_j \\ c_{j+1} \end{pmatrix} = \begin{pmatrix} f_j^q \\ f_{j+1}^q \end{pmatrix} = \boldsymbol{H}_2 \begin{pmatrix} f_j^{q-1} \\ f_{j+1}^{q-1} \end{pmatrix} = \begin{pmatrix} f_j^{q-1} + f_{j+1}^{q-1} \\ f_j^{q-1} - f_{j+1}^{q-1} \end{pmatrix}$$

$$j = 0,2,4,\ldots,M-2 \, ; \quad M = 2^q \, . \tag{3.2.65}$$

Die WHT gemäß (3.2.59) erfordert $\mathcal{O}(M^2)$ Additionen und Subtraktionen, die WHT gemäß Bild 3.2.7 nur $\mathcal{O}(M \mathrm{ld} M)$. Die Verallgemeinerung auf die Zerlegung einer Matrix $\boldsymbol{H}_M$, $M > 8$, $M = 2^q$ dürfte offensichtlich sein und ist auch in (3.5.1) angegeben. Mehrdimensionale Transformationen lassen sich, wie schon in Abschnitt 2.3.3 für die DFT ausgeführt wurde, auf mehrere eindimensionale zurückführen. Weitere ähnliche Transformationen sind in der zitierten Literatur enthalten.

3.2.6 HAAR-Transformation

Als letztes Beispiel dient die HAAR-Transformation, die bis auf eine Skalierung ebenfalls nur Funktionswerte 0 oder 1 verwendet, jedoch eine orts– bzw. zeitabhängige Komponente einführt.

Definition 3.6 *Die* HAAR-**Funktionen** $h_k(x) = h_{pq}(x)$ *werden im Intervall* $0 \le x \le 1$ *definiert für Indizes*

$$\begin{aligned} k &= 0,1,\ldots,N-1 \, , \quad N = 2^n \, , \\ k &= 2^p + q - 1 \, , \quad 0 \le p \le n-1 \, , \\ q &= \begin{cases} 0,1 & : \; p = 0 \, , \\ 1,2,\ldots,2^p & : \; p \ne 0 \, . \end{cases} \end{aligned} \tag{3.2.66}$$

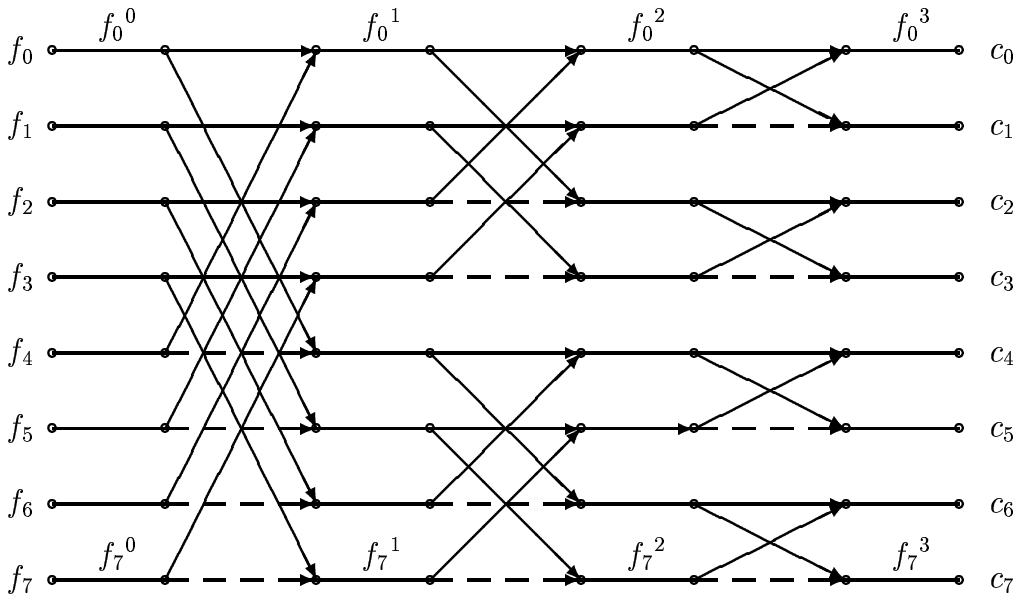


Bild 3.2.7: Der Signalflussgraph der schnellen HADAMARD geordneten WALSH–HADAMARD-Transformation für $M = 8$

Sie sind gegeben durch die Gleichungen

$$
\begin{aligned}
h_{00}(x) &= \frac{1}{\sqrt{N}}\,, \\
h_{pq}(x) &= \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & : \ \dfrac{q-1}{2^p} \le x \le \dfrac{q-\frac{1}{2}}{2^p}\,, \\ -2^{p/2} & : \ \dfrac{q-\frac{1}{2}}{2^p} \le x \le \dfrac{q}{2^p}\,, \\ 0 & : \ \text{sonst für } x \in [0,1]\,. \end{cases}
\end{aligned}
\tag{3.2.67}
$$

Beispiele von HAAR-Funktionen für $p = 0, 1, 2$ zeigt Bild 3.2.8, in dem die Funktionswerte alle auf 1 normiert sind,

Die *diskrete* HAAR-**Transformation** (HRT) erhält man, indem man diskrete Werte von x an den Stellen $\frac{m}{N},\ m = 0, 1, \ldots, N-1$ betrachtet. Sie ist auch die Basis für eine spezielle Wavelet Transformation, wie in Abschnitt 3.3 noch erläutert wird. Für $N = 8$ ergibt sich so aus der obigen Definition die diskrete Transformation

$$
\mathrm{HRT}(\boldsymbol{f}) = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix} \cdot \boldsymbol{f} \tag{3.2.68}
$$

p	q	k	h_{pq}
0	0	0	h_{00}
0	1	1	h_{01}
1	1	2	h_{11}
1	2	3	h_{12}
2	1	4	h_{21}
2	2	5	h_{22}
2	3	6	h_{23}
2	4	7	h_{24}

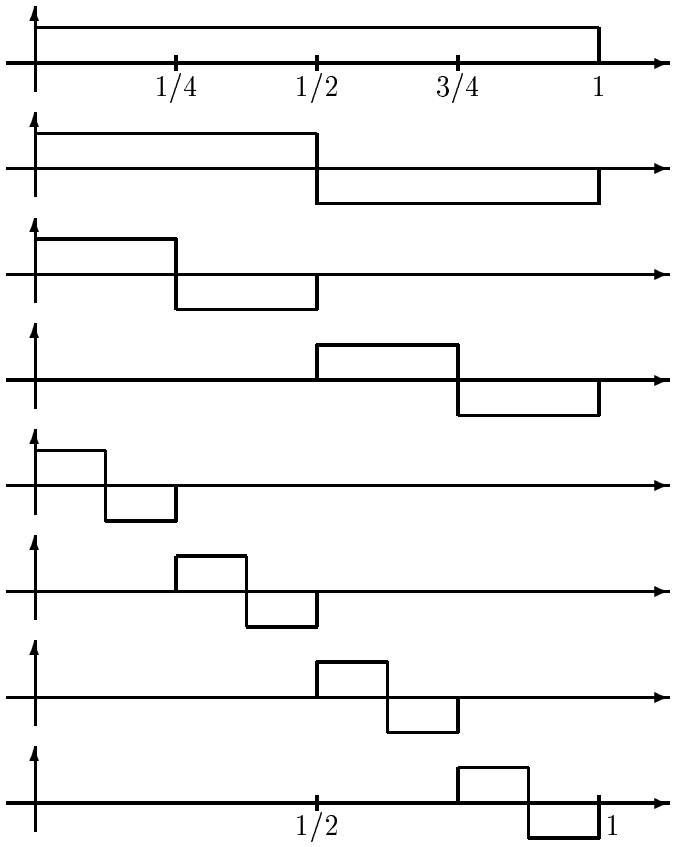


Bild 3.2.8: Einige HAAR-Funktionen

Während die Funktionen von Transformationen wie FOURIER oder WALSH stets im gesamten Definitionsbereich von Null verschiedene Werte hatten, wird der Definitionsbereich der HAAR-Funktionen also auf immer kleinere Bereiche eingeschränkt.

3.3 Wavelet–Transformation (VA.2.3.2, 31.10.2005)

3.3.1 Kontinuierliche Wavelet–Transformation

Bei der Wavelet–Transformation wird eine Basisfunktion $\psi(t)$, die als "Wavelet" („kleine Welle" oder „Wellchen") bezeichnet wird, nach einer Skalierung um den Faktor α an die Position τ der Zeitachse geschoben und dann eine Integraltransformation berechnet. Damit erhält man eine Darstellung einer Funktion $f(t)$, die sowohl von der Position τ als auch von der Frequenz α abhängt, während die FOURIER-Transformierte nur von der Frequenz abhängt. Es wird vorausgesetzt, dass alle verwendeten Funktionen quadratisch integrierbar sind im Sinne von (3.2.7).

Definition 3.7 *Die* **Wavelet–Transformation** *(*WT*) und ihre Inverse sind definiert durch*

$$\boxed{\begin{aligned} \mathrm{WT}(\tau,\alpha) &= \int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt{|\alpha|}}\psi^*\left(\frac{t-\tau}{\alpha}\right)\,\mathrm{d}t = \int_{-\infty}^{\infty} f(t)\psi_{\alpha,\tau}(t)\,\mathrm{d}t\;, \\ f(t) &= \frac{1}{\alpha^2 c_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mathrm{WT}(\tau,\alpha)\frac{1}{\sqrt{|\alpha|}}\psi\left(\frac{t-\tau}{\alpha}\right)\,\mathrm{d}\tau\,\mathrm{d}\alpha\;. \end{aligned}} \tag{3.3.1}$$

Eine Funktion $\psi(t)$ ist als Basisfunktion *zulässig*, wenn für ihre FOURIER-Transformierte $\Psi(\omega)$ gilt

$$c_\psi = \int_{-\infty}^{\infty}\frac{|\Psi(\omega)|^2}{|\omega|}\,\mathrm{d}\omega < \infty\;. \tag{3.3.2}$$

Diese Bedingung ist *notwendig* für die Existenz der inversen Transformation. Sie kann nur erfüllt sein, wenn $\Psi(0) = 0$, d. h. wenn das Wavelet *keinen Gleichanteil* besitzt. Die Bedingung erfordert auch, dass die Basisfunktion genügend rasch gegen Null geht, und daher kommt die Bezeichnung Wavelet.

Ein Beispiel für eine zulässige Basisfunktion ist das MORLET-**Wavelet** in (3.3.3), das in Bild 3.3.1 gezeigt ist, sowie die Funktion in (3.3.4), welche bis auf Faktoren (2.3.59), S. 104, entspricht und die zweite Ableitung einer GAUSS-Funktion ist, die in Bild 2.3.10, S. 105, gezeigt ist,

$$\psi_{\mathrm{M}}(t) = \exp\left[-\alpha t^2\right]\,\exp[\mathrm{i}\,\omega t]\;,\quad \alpha > 0\;, \tag{3.3.3}$$

$$\psi_{\mathrm{G}}(t) = (1-t^2)\exp\left[-\frac{t^2}{2}\right]\;. \tag{3.3.4}$$

3.3.2 Wavelet Reihe

Die Werte von α, τ werden nun auf die diskreten Werte

$$\alpha = 2^{-\mu}\;,\quad \tau = k\alpha\;,\quad \mu, k = \ldots, 0, \pm1, \pm2, \ldots \tag{3.3.5}$$

eingeschränkt (in der Literatur findet man auch die Definition $\alpha = 2^{\mu}$). Damit wird eine Familie von Wavelets $\{\psi_{\mu,k}(t)\}$ aus einer Basisfunktion $\psi(t)$ durch *Normierung*, *Skalierung* und *Verschiebung* generiert

$$\psi_{\mu,k}(t) = \frac{1}{\sqrt{|\alpha|}}\psi\left(\frac{t-\tau}{\alpha}\right) = \sqrt{2^{\mu}}\,\psi\left(2^{\mu}t - k\right)\;,\quad \psi_{0,0}(t) = \psi(t)\;. \tag{3.3.6}$$

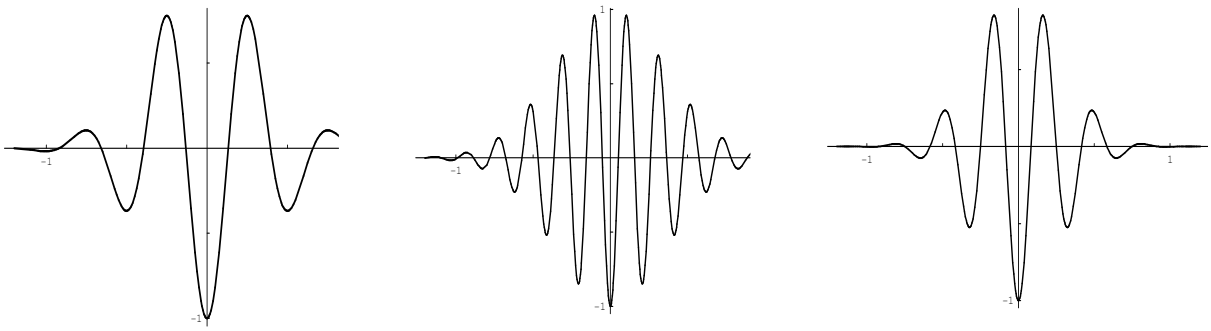


Bild 3.3.1: Einige MORLET-Wavelets (dargestellt ist der Betrag); von links nach rechts: $(\alpha;\omega) = (3,8;\ 2\pi 1,9), (3,8;\ 2\pi 4,8), (6;\ 2\pi 3,0)$

Man nennt die Familie von Wavelets *orthonormal*, wenn sie die Bedingung erfüllt

$$\int_{-\infty}^{\infty} \psi_{\mu,k}(t)\psi^*_{\nu,l}(t)\,\mathrm{d}t = \begin{cases} 1 & : \quad \mu = \nu \quad \text{und} \quad k = l\ , \\ 0 & : \quad \text{sonst}\ . \end{cases} \tag{3.3.7}$$

Definition 3.8 *Die* **Wavelet Reihe** *einer Funktion $f(t)$ ist definiert durch*

$$\boxed{\begin{aligned} f(t) &= \sum_{\mu=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{\mu,k}\psi_{\mu,k}(t)\ , \\ d_{\mu,k} &= \int_{-\infty}^{\infty} f(t)\psi_{\mu,k}(t)\,\mathrm{d}t = \langle f(t), \psi_{\mu,k}(t)\rangle\ . \end{aligned}} \tag{3.3.8}$$

Die d_{jk} sind die **Wavelet Koeffizienten**.

Ein Vergleich mit (3.2.6), S. 168, zeigt, dass die Wavelet Reihe einfach eine spezielle orthonormale Reihenentwicklung ist. Die Wavelet Reihe ist, ähnlich wie die FOURIER-Reihe, eine unendliche Reihenentwicklung und erfordert damit bei numerischen Rechnungen den Abbruch nach endlich vielen Termen. Mit dem Konzept der *Auflösungshierarchie* bzw. der Multiresolutionsdarstellung oder Darstellung in Auflösungsstufen ist eine definierte Auswahl gewünschter Detailstufen möglich.

3.3.3 Auflösungshierarchie

Teilbandkodierung

Die Wavelet Transformation in (3.3.1) lässt sich umformen in

$$\mathrm{WT}(\tau,\alpha) = \sqrt{|\alpha|}\int_{-\infty}^{\infty} f(\alpha t)\psi^*\left(t - \frac{\tau}{\alpha}\right)\,\mathrm{d}t\ . \tag{3.3.9}$$

Daran sieht man, dass die Funktion $f(\alpha t)$ bei Vergrößerung des *Maßstabs* $\alpha > 0$ zunehmend *komprimiert* und bei Verkleinerung des Maßstabs zunehmend *expandiert* wird. Ein großer Maßstab entspricht einer globalen Sicht auf f. Ein etwas anderer Begriff ist die *Auflösung*, mit der

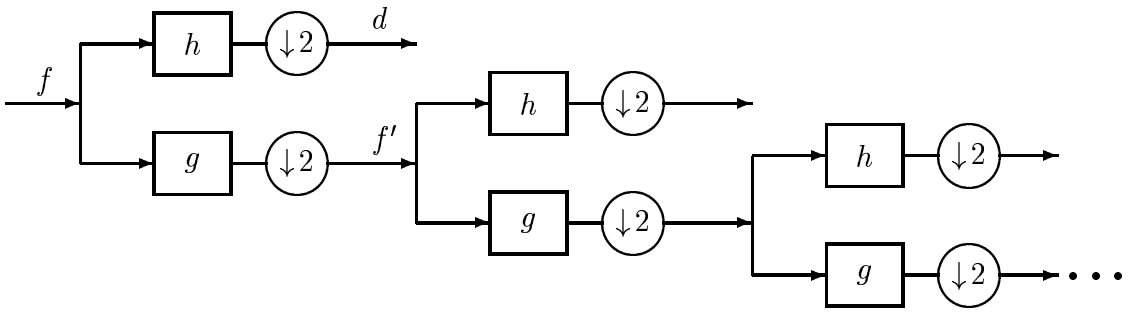


Bild 3.3.2: Eine Auflösungshierarchie kann durch wiederholte Tief– und Hochpassfilterung berechnet werden

die Funktion dargestellt wird. Diese hängt ab vom Umfang des Frequenzbandes. Durch eine Tiefpassfilterung wird das Frequenzband reduziert, die Auflösung verringert, aber der Maßstab erhalten. Durch eine Kompression der Funktion wird die Auflösung erhalten, da die Kompression reversibel ist, aber der Maßstab vergrößert. Dieses gilt für *kontinuierliche* Funktionen $f(t)$, jedoch *nicht* für die Abtastwerte einer diskreten Funktion f_j, die unter Beachtung des Abtasttheorems aus einer kontinuierlichen Funktion gewonnen wurde. Die *Vergrößerung* des Maßstabs einer diskreten Funktion erfordert eine Unterabtastung, und damit wird die Auflösung *reduziert*.

Bei einer diskreten Funktion f_j kann man eine **Auflösungshierarchie** durch die Vorgehensweise der *Teilbandkodierung* berechnen. Dabei wird, wie in Bild 3.3.2 gezeigt, das Frequenzband durch ein ideales Tiefpassfilter g und ein ideales Hochpassfilter h in je zwei Hälften zerlegt, die die Signale f' und d ergeben. Die Tief– und Hochpassfilterung kann erneut auf f' angewendet werden, usw. Damit entsteht eine Folge von Funktionen abnehmender Auflösung. Die Rekonstruktion des Originals f ist zwar aus f' nicht fehlerfrei möglich, aber natürlich aus der Kombination von f' und d, da diese die vollständige Information enthalten. Das Ergebnis f' der Tiefpassfilterung ist eine Art „Mittelwertsignal“, das Ergebnis d der Hochpassfilterung ein „Differenzsignal“.

Skalierungsfunktion und Wavelet

Die oben erwähnten idealen Filter sind ein Spezialfall, der nur näherungsweise realisierbar ist. Die Theorie der Wavelet–Transformation hat gezeigt, dass Auflösungshierarchieen unter sehr viel allgemeineren Bedingungen berechenbar sind. Diese beruht auf der Definition einer geeigneten **Skalierungsfunktion** ϕ, mit der die Auflösung der Funktion f zunehmend reduziert wird, sowie einem dazu passenden Wavelet ψ, mit dem die Differenzinformation berechnet wird. Die Ergebnisse der Transformationen mit ϕ und ψ erlauben dann wieder die Rekonstruktion von f. Skalierungsfunktion und Wavelet übernehmen also in verallgemeinerter Form die Rolle von idealem Tief– und Hochpass.

Wir fassen im Folgenden einige wichtige Ergebnisse zur Existenz entsprechender Skalierungsfunktionen und Wavelets zusammen und verweisen für die Beweise auf die in Abschnitt 3.11 zitierte Literatur. Es lässt sich zeigen, dass es Skalierungsfunktionen $\phi(t)$ gibt sowie

dazugehörige orthogonale Wavelets $\psi(t)$. Weiter wurde gezeigt, dass

$$\begin{aligned} \phi_{\mu,k}(t) &= \sqrt{2^\mu}\,\phi\left(2^\mu t - k\right) , \\ \psi_{\mu,k}(t) &= \sqrt{2^\mu}\,\psi\left(2^\mu t - k\right) \end{aligned} \tag{3.3.10}$$

für $\mu, k \in \mathsf{Z}$ Basisfunktionen des $\mathsf{L}^2(\mathbb{R})$ sind. Für $k \in \mathsf{Z}$ sind die $\phi_{\mu,k}(t)$ Basisfunktionen eines Unterraumes $V_\mu \subset \mathsf{L}^2(\mathbb{R})$ und die $\psi_{\mu,k}(t)$ Basisfunktionen eines dazu orthogonalen Unterraumes $W_\mu \subset \mathsf{L}^2(\mathbb{R})$. Wir beschränken uns hier auf orthogonale Skalierungs– und Waveletfunktionen und verweisen für allgemeinere, wie z. B. biorthogonale Wavelets, auf die Literatur. Es gilt also

$$0 = \int \phi_{\mu,k}(t)\psi_{\mu,l}(t)\,\mathrm{d}t = \langle \phi_{\mu,k}(t)\psi_{\mu,l}(t) \rangle \ . \tag{3.3.11}$$

Diese Unterräume erfüllen die **Hierarchiebedingungen** (oder die *Multiresolutionsbedingungen*) Hier werden nur zwei einfache Spezialfälle erwähnt:

1. *Schachtelung* der Unterräume

$$\begin{aligned} V_\mu &\subset V_{\mu+1} , \qquad f(t) \in V_\mu \Longleftrightarrow f(2t) \in V_{\mu+1} , \\ W_\mu &\subset W_{\mu+1} , \qquad f(t) \in W_\mu \Longleftrightarrow f(2t) \in W_{\mu+1} , \end{aligned} \tag{3.3.12}$$

2. *direkete Summe* von Skalierungs– und Waveletanteil (bzw. von Mittelwert– und Differenzanteil)

$$\begin{aligned} V_1 &= V_0 \oplus W_0 , \quad V_2 = V_0 \oplus W_0 \oplus W_1 , \quad \ldots \\ V_{\mu+1} &= V_\mu \oplus W_\mu , \quad V_\mu \perp W_\mu , \end{aligned} \tag{3.3.13}$$

3. *Vollständigkeit* der Basisfunktionen

$$\begin{aligned} \{\emptyset\} &= V_{-\infty} \subset \ldots \subset V_0 \subset V_1 \subset \ldots \subset V_\infty = \mathsf{L}^2(\mathbb{R}) , \\ V_\infty &= V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_\infty = \mathsf{L}^2(\mathbb{R}) . \end{aligned} \tag{3.3.14}$$

Der Index μ nimmt also im Prinzip Werte zwischen $-\infty, \ldots, +\infty$ an, wobei ein *größerer* Wert des Index einer *feineren* Auflösung entspricht. Aus den Hierarchiebedingungen, insbesondere aus (3.3.14) folgt, dass sich die Entwicklung einer Funktion mit Hinzunahme der Skalierungsfunktion bei einem beliebigen Index μ, z. B. $\mu = 0$, beginnen lässt, sodass sich die Waveletreihe (3.3.8) modifiziert zu

$$\boxed{f(t) = \sum_k f_{0,k}\phi_{0,k}(t) + \sum_{\mu=0}^{\infty}\sum_k d_{\mu,k}\psi_{\mu,k}(t)} \tag{3.3.15}$$

Praktisch geht man natürlich immer von Mustern bzw. Signalen *begrenzter* Auflösung aus, da technische Sensoren eine begrenzte Bandbreite haben und ihre Messwerte für die digitale Verarbeitung nach den Grundsätzen von Abschnitt 2.1.2 abgetastet werden. Daher gibt es eine durch den Sensor bedingte feinste Auflösungsstufe $\mu = m$, wenn man $M = 2^m$ Abtastwerte der Funktion $f(t)$ aufnimmt, sowie eine gröbste Auflösung $\mu = 0$. Natürlich ist mit dieser begrenzten Auflösung $\mathsf{L}^2(\mathbb{R})$ nicht mehr exakt darstellbar.

Schließlich lässt sich zeigen, dass eine mehrdimensionale Erweiterung in einfacher Weise möglich ist, wenn man von einer *separierbaren* Darstellung ausgeht. Ist nämlich $\phi(t)$ eine Skalierungsfunktion mit obigen Eigenschaften, so ist

$$\phi_{\mu,\nu;k,l}(x,y) \;=\; \sqrt{2^\mu}\,\phi\left(2^\mu x - k\right)\sqrt{2^\nu}\,\phi\left(2^\nu y - l\right) \tag{3.3.16}$$

für $\mu, \nu, k, l \in \mathsf{Z}$ eine Basis des $\mathsf{L}^2(\mathbb{R}^2)$ der zweidimensionalen quadratisch integrierbaren Funktionen $f(x,y)$, bzw. für $k, l \in \mathsf{Z}$ eine Basis eines Unterraumes $V^2_{\mu,\nu} \subset \mathsf{L}^2(\mathbb{R}^2)$. Analog ergibt sich aus

$$\begin{aligned}
\psi_0(x,y) &= \phi(x)\,\psi(y)\;, \\
\psi_1(x,y) &= \psi(x)\,\psi(y)\;, \\
\psi_2(x,y) &= \psi(x)\,\phi(y)
\end{aligned} \tag{3.3.17}$$

eine Basis eines Unterraumes $W^2_{\mu,\nu} \subset \mathsf{L}^2(\mathbb{R}^2)$. Die Unterräume erfüllen eine zu (3.3.13) analoge Hierarchiebedingung. Im Folgenden werden daher nur eindimensionale Wavelets weiter betrachtet.

Wegen (3.3.12) ist jede Funktion, die ein Element von V_μ ist, auch ein Element von $V_{\mu+1}$. Daraus folgt, dass sich die Basisfunktionen $\phi_{\mu,k}$ durch Basisfunktionen $\phi_{\mu+1,k}$ darstellen lassen mit (3.2.6), S. 168

$$\begin{aligned}
\phi_{\mu,k}(t) &= \sum_l \langle \phi_{\mu,k}(\tau), \phi_{\mu+1,l}(\tau)\rangle\, \phi_{\mu+1,l}(t) \\
&= \sum_l \left(\sqrt{2}\int_{-\infty}^{\infty} \phi(u)\phi\left(2u-(l-2k)\right)\,\mathrm{d}u\right)\phi_{\mu+1,l}(t)\;, \qquad (3.3.18)\\
\phi_{\mu,k}(t) &= \sum_l g_{l-2k}\phi_{\mu+1,l}(t)\;, \qquad (3.3.19)\\
g_{l-2k} &= \sqrt{2}\int \phi(u)\phi\left(2u-(l-2k)\right)\,\mathrm{d}u\;. \qquad (3.3.20)
\end{aligned}$$

Die obigen Umformungen ergeben sich durch die Variablensubstitution $2^\mu\tau - k = u$ im inneren Produkt und Einführung der Entwicklungskoeffizienten g_{l-2k}. Mit $k = \mu = 0$ und (3.3.10) erhält man daraus die *Verfeinerungsgleichung* oder **Zweiskalengleichung**

$$\boxed{\begin{aligned}
\phi(t) &= \sqrt{2}\sum_l g_l\,\phi(2t-l)\;, \\
g_l &= \sqrt{2}\int_{-\infty}^{\infty}\phi(u)\phi(2u-l)\,\mathrm{d}u = \langle\phi_{0,0}(u),\phi_{1,l}(u)\rangle\;.
\end{aligned}} \tag{3.3.21}$$

Aus (3.3.13) folgt, dass auch die $\psi_{\mu,k}$ sich durch $\phi_{\mu+1,k}$ darstellen lassen. Eine analoge Rechnung ergibt für die Wavelets dann die (3.3.19) entsprechende Gleichung

$$\psi_{\mu,k}(t) \;=\; \sum_l h_{l-2k}\phi_{\mu+1,l}(t)\;. \tag{3.3.22}$$

Aus dieser ergibt sich die analoge Zweiskalengleichung; aus der Orthonormalität der Räume V_μ

und W_μ ergibt sich zudem eine Beziehung zwischen den Koeffizienten g_l und h_l zu

$$\boxed{\begin{aligned} \psi(t) &= \sqrt{2}\sum_l h_l\,\phi(2t-l)\;, \\ h_l &= \sqrt{2}\int_{-\infty}^{\infty}\psi(u)\phi(2u-l)\,\mathrm{d}u = \langle\psi_{0,0}(u),\phi_{1,l}(u)\rangle\;, \\ h_l &= (-1)^l g_{1-l}\;. \end{aligned}} \tag{3.3.23}$$

Ein einfaches Beispiel für eine Skalierungsfunktion und das zugehörige Wavelet sind die HAAR-Funktionen aus Abschnitt 3.2.6. Wählt man eine Skalierungsfunktion, die im Intervall $0 \le t \le 1$ definiert ist, so folgt aus (3.3.21)

$$\begin{aligned} \phi(t) &= \begin{cases} 1 & : \; 0 < t < 1 \\ 0 & : \; \text{sonst} \end{cases}\;, \\ g_0 &= g_1 = \frac{1}{\sqrt{2}}\;. \end{aligned} \tag{3.3.24}$$

Für das Wavelet ergibt sich aus (3.3.23)

$$\begin{aligned} \psi(t) &= \begin{cases} 1 & : \; 0 < t < 0,5 \\ -1 & : \; 0,5 < t < 1 \\ 0 & : \; \text{sonst} \end{cases}\;, \\ h_0 &= \frac{1}{\sqrt{2}}\;, \quad h_1 = -\frac{1}{\sqrt{2}}\;. \end{aligned} \tag{3.3.25}$$

Bild 3.3.3 zeigt die Funktionen $\phi_{3,j}$, $\phi_{2,j}$, $\psi_{2,j}$. Ein Vergleich mit der HAAR-Transformation und Bild 3.2.8 zeigt, dass dort offenbar die hochfrequenten Anteile weiter zerlegt werden, hier die niederfrequenten. Die Aufnahme eines Bildes mit einer CCD–Kamera kann näherungsweise als Entwicklung mit HAAR-Funktionen $\phi(x)\phi(y)$ in der feinsten Auflösungsstufe aufgefasst werden. Das Abtasttheorem, Satz 2.1, S. 65, geht ja von einer Abtastung an einem Zeit*punkt* aus, was technisch nur näherungsweise realisierbar ist. Auf eine genauere Analyse dieser Problematik kann hier jedoch verzichtet werden.

Pyramidenalgorithmus für die Koeffizientenberechnung

Mit den Basisfunktionen $\phi_{\mu,k}, \psi_{\mu,k}$ aus (3.3.10) wird eine Funktion $f(t)$ in die Unterräume V_μ, W_μ der Auflösungsstufe μ projiziert bzw. nach den Basisfunktionen dieser Unterräume entwickelt. Wir bezeichnen die Projektionsoperationen mit Φ_μ, Ψ_μ. Für Φ_μ ergibt sich

$$\begin{aligned} \Phi_\mu\{f(t)\} &= \sum_k f_{\mu,k}\sqrt{2^\mu}\phi\left(2^\mu t-k\right) = \sum_k f_{\mu,k}\phi_{\mu,k} \in V_\mu\;, \\ f_{\mu,k} &= \int_{-\infty}^{\infty} f(\tau)\phi_{\mu,k}(\tau)\,\mathrm{d}\tau = \langle f(\tau),\phi_{\mu,k}(\tau)\rangle \end{aligned} \tag{3.3.26}$$

und entsprechend für Ψ_μ

$$\Psi_\mu\{f(t)\} = \sum_k d_{\mu,k}\sqrt{2^\mu}\psi\left(2^\mu t-k\right) = \sum_k d_{\mu,k}\psi_{\mu,k} \in W_\mu\;,$$

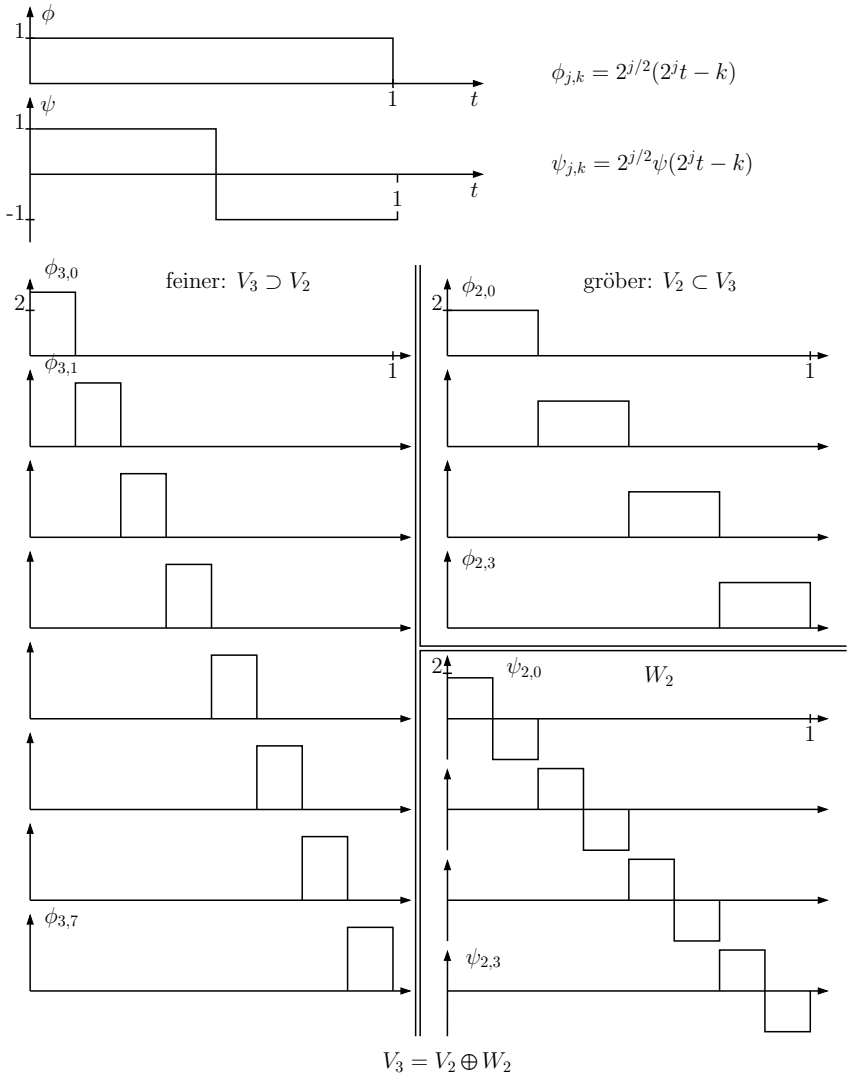


Bild 3.3.3: Die HAAR-Funktionen $\phi(t)$ und $\psi(t)$ sowie $\phi_{3,j},\ \phi_{2,j},\ \psi_{2,j}$ als Beispiele für einfache Skalierungsfunktionen und Wavelets.

$$d_{\mu,k} \quad = \quad \int_{-\infty}^{\infty} f(\tau)\psi_{\mu,k}(\tau)\,\mathrm{d}\tau = \langle f(\tau), \psi_{\mu,k}(\tau)\rangle \ . \tag{3.3.27}$$

Damit kann eine Funktion $f(t)$ ausgehend von der anfänglich gegebenen Auflösung $\mu = m$ mit von Schritt zu Schritt abnehmender Auflösung dargestellt werden durch die Koeffizienten $f_{\mu,k}$; die Differenz zur vorangehenden Auflösung $\mu + 1$ ergibt sich wegen (3.3.13) aus den Koeffizienten $d_{\mu,k}$. Die Bezeichnung $f_{\mu,k}$ für den ersten Koeffizienten soll andeuten, dass es sich um eine gröbere Darstellung des Ausgangssignals handelt, die Bezeichnung $d_{\mu,k}$, dass hier die Differenz– oder Detailinformation enthalten ist. Es gilt also wegen (3.3.13) für eine Funktion $f(t) \in V_{\mu+1}$

$$\Phi_{\mu+1}\{f(t)\} \quad = \quad \Phi_\mu\{f(t)\} + \Psi_\mu\{f(t)\} \ . \tag{3.3.28}$$

Mit (3.3.28) und (3.3.26), (3.3.27) erhält man für eine Funktion $f(t) \in V_{\mu+1}$

$$\begin{aligned} f(t) &= \sum_k f_{\mu+1,k}\phi_{\mu+1,k}(t) \\ &= \sum_k f_{\mu,k}\phi_{\mu,k}(t) + \sum_k d_{\mu,k}\psi_{\mu,k}(t) \ . \end{aligned} \tag{3.3.29}$$

Durch Verwendung von (3.3.26) und Einsetzen von (3.3.19) ergibt sich daraus eine Beziehung zwischen den Entwicklungskoeffizienten der feineren Stufe $\mu + 1$ und der gröberen Stufe μ zu

$$\begin{aligned} f_{\mu,k} &= \int f(t)\phi_{\mu,k}(t)\,\mathrm{d}t \\ &= \int f(t)\sum_l g_{l-2k}\phi_{\mu+1,l}(t)\,\mathrm{d}t \\ &= \sum_l g_{l-2k}\int f(t)\phi_{\mu+1,l}(t)\,\mathrm{d}t \\ &= \sum_l g_{l-2k} f_{\mu+1,l} \ . \end{aligned} \tag{3.3.30}$$

Eine entsprechende Rechnung für die $d_{\mu,k}$ ergibt die beiden Gleichungen des **Pyramidenalgorithmus** bzw. die *Analysegleichungen*

$$\boxed{f_{\mu,k} = \sum_l g_{l-2k} f_{\mu+1,l} \ , \quad \text{und} \quad d_{\mu,k} = \sum_l h_{l-2k} f_{\mu+1,l} \ .} \tag{3.3.31}$$

Damit kann man, ausgehend von einer anfänglich gegebenen feinsten Auflösung $\mu = m$, schrittweise die Entwicklungskoeffizienten $f_{\mu,k}$, $d_{\mu,k}$ der nächst gröberen Auflösungsstufen berechnen, *ohne* ständig die inneren Produkte in (3.3.26), (3.3.27) auszuwerten, sodass eine *effiziente Berechnung* der Wavelet–Transformation möglich ist. Die Auflösung $\mu = m$ wird dabei in der Regel den anfänglichen Abtastwerten entsprechen. Wenn die Abtastrate hoch genug ist (s. Satz 2.1, S. 65), sind die Abtastwerte f_j eine gute Approximation an die Koeffizienten $f_{m,j}$ in (3.3.31), d. h. in der anfänglichen feinsten Auflösung.

Aus den Koeffizienten lässt sich die Funktion in der ursprünglichen Auflösung rekonstruieren (bzw. synthetisieren). Man geht dazu von (3.3.29) aus

$$\sum_k f_{\mu+1,k}\phi_{\mu+1,k}(t) = \sum_k f_{\mu,k}\phi_{\mu,k}(t) + \sum_k d_{\mu,k}\psi_{\mu,k}(t) \ , \tag{3.3.32}$$

setzt dort (3.3.19) und (3.3.22) ein und erhält

$$\sum_k f_{\mu+1,k}\phi_{\mu+1,k}(t) = \sum_k f_{\mu,k}\sum_l g_{l-2k}\phi_{\mu+1,l}(t) + \sum_k d_{\mu,k}\sum_l h_{l-2k}\phi_{\mu+1,l}(t) \ .$$

Multipliziert man linke und rechte Seite mit $\phi_{\mu+1,\lambda}(t)$ und integriert nach t, so folgt wegen der Orthonormalität der Skalierungsfunktionen die *Synthesegleichung*

$$\boxed{f_{\mu+1,k} = \sum_l f_{\mu,l} g_{k-2l} + \sum_l d_{\mu,l} h_{k-2l}} \tag{3.3.33}$$

Diese Gleichung gibt den Übergang von gröberen zu feineren Stufen.

3.3.4 Diskrete Wavelet Transformation einer endlichen Folge

Die obigen Summen wurden in der Regel nur mit $\sum_l$ angegeben, die Grenzen offengelassen. Sie reichen i. Allg. von $(-\infty, \infty)$, was natürlich für die digitale Verarbeitung, insbesondere die digitale Verarbeitung einer endlichen Folge von Abtastwerten, unzweckmäßig ist. Wir geben daher noch kurz die Transformation einer endlichen Zahl von Abtastwerten an.

Die Ausgangsgleichungen sind nach wie vor (3.3.8) bzw. (3.3.15), wobei letztere zeigt, dass man die Entwicklung bei einem *beliebigen* Index $\mu = \mu_0$, z. B. $\mu = \mu_0 = 0$, beginnen kann, indem man die Skalierungsfunktion $\phi_{\mu_0,k}(t)$ hinzunimmt. Das ergibt

$$\begin{aligned} f(t) &= \sum_{k=-\infty}^{\infty} f_{\mu_0,k}\phi_{\mu_0,k}(t) + \sum_{\mu=\mu_0}^{\infty}\sum_{k=-\infty}^{\infty} d_{\mu,k}\psi_{\mu,k}(t) \\ &= \sum_{k}\langle f, \phi_{\mu_0,k}\rangle\phi_{\mu_0,k}(t) + \sum_{\mu=\mu_0}^{\infty}\sum_{k}\langle f, \psi_{\mu,k}\rangle\psi_{\mu,k}(t)\ . \end{aligned} \tag{3.3.34}$$

Damit ist der untere Index der Summe über μ bereits endlich gemacht. Der Index $\mu = \mu_0$ gibt die *gröbste* Auflösungsstufe an.

Die Wahl eines endlichen oberen Index $\mu = m$ für die Skalierungsfunktion impliziert i. Allg. eine Approximation. Da man in der digitalen Verarbeitung $f(t)$ durch seine Abtastwerte f_j repräsentiert, ist ohnehin eine Bandbegrenzung von $f(t)$ vorausgesetzt, die in der Regel durch Tiefpassfilterung sichergestellt wird; die tiefpassgefilterte Funktion sei $f_m(t)$. Damit ergibt die Rekonstruktion der (tiefpassgefilterten) Funktion $f_m(t)$ aus den Abtastwerten mit Hilfe der Interpolationsformel (2.1.21), S. 65, auch nur eine Approximation der ursprünglichen (noch nicht tiefpassgefilterten) Funktion $f(t)$. Zur Berechnung der Wavelet Koeffizienten von $f_m(t)$ aus den Abtastwerten f_j müssen, ähnlich wie bei der diskreten FOURIER-Transformation, die Abtastwerte dicht genug gewält sein. Die feinste Auflösungsstufe, oberhalb derer fast keine Signalenergie mehr liegt, sei $\mu = m$, wobei wir hier der Einfachheit halber voraussetzen, dass die Zahl der Abtastwerte $M = 2^m$ ist. Das bedeutet, dass mit guter Annäherung $f(t) \in V_m$ gelten sollte, bzw. $\Phi_m\{f(t)\} \approx f(t)$. Im Folgenden nehmen wir an, dass diese Forderung exakt erfüllt ist, d. h.

$$f(t) = \sum_{k=-\infty}^{\infty} f_{\mu_0,k}\phi_{\mu_0,k}(t) + \sum_{\mu=\mu_0}^{m-1}\sum_{k=-\infty}^{\infty} d_{\mu,k}\psi_{\mu,k}(t)\ . \tag{3.3.35}$$

Die sukzessive Reduktion der Auflösung zeigt Bild 3.3.4 an einem schematisierten Beispiel.

Die Berechnung der DWT erfordert dann die Schritte

1. Projektion von $f(t)$ auf die feinste Auflösungsstufe,

$$\begin{aligned} f_{m,k} &= \Phi_m\{f(t)\} = \langle f, \phi_{m,k}\rangle \\ &\approx \sum_{l} f(2^{-m}l)\phi_{m,l-k}\ . \end{aligned} \tag{3.3.36}$$

2. Berechnung der Wavelet Koeffizienten $d_{\mu,k} = \langle f, \psi_{\mu,k}\rangle,\ \mu = m-1, m-2, \ldots, \mu_0$ und $f_{\mu_0,k} = \langle f, \phi_{\mu_0,k}\rangle$ mit dem Pyramidenalgorithmus (3.3.31).

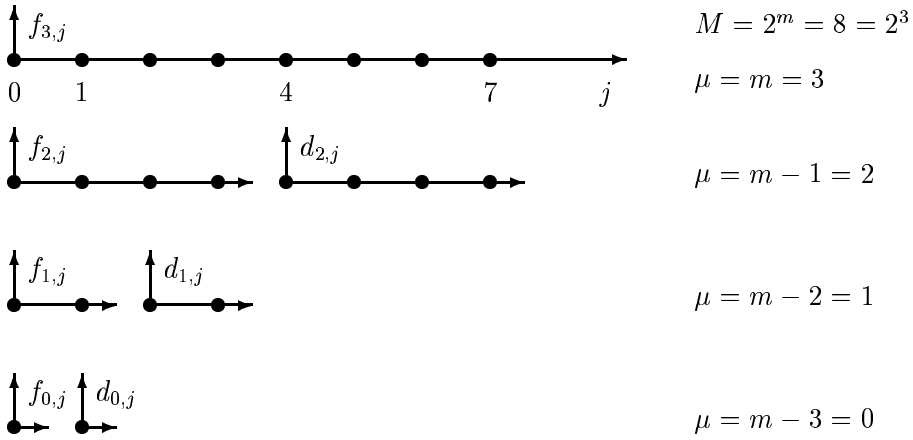


Bild 3.3.4: Auflösungsreduktion durch die Wavelet Transformation am Beispiel $M = 2^3$

In der zitierten Literatur wird erwähnt, dass praktisch die Abtastrate die feinste Auflösungsstufe bestimmt und dass oft statt (3.3.36) einfach die Abtastwerte selbst als Koeffizienten auf der feinsten Stufe verwendet werden, d. h. man setzt oft $f_{m,k} = f_j$ (mit $f_{m,k}$ wie üblich die Wavelet Koeffizienten auf der Auflösungsstufe m, f_j die ursprünglichen Abtastwerte).

Wenn insbesondere $f(t)$ nur in einem endlichen Intervall $0 \le t \le T$ von Null verschiedene Werte annimmt, wird analog wie bei der diskreten FOURIER-Transformation ein periodisches Signal

$$\widetilde{f}(t) = \sum_l f(t + lT) \tag{3.3.37}$$

konstruiert, wobei T die ganzzahlige Periodenlänge ist. Es lässt sich zeigen, dass für $\widetilde{f}(t)$ die Koeffizienten $\widetilde{f}_{\mu,k}$, $\widetilde{d}_{\mu,k}$ der Skala μ ebenfalls periodisch sind mit Periodenlänge $2^\mu T$

$$\widetilde{f}(t) = \widetilde{f}(t + T) \iff \widetilde{f}_{\mu,k} = \widetilde{f}_{\mu,(k+2^\mu T)} \,,\; \widetilde{d}_{\mu,k} = \widetilde{d}_{\mu,(k+2^\mu T)} \tag{3.3.38}$$

und sich berechnen aus

$$\begin{aligned} \widetilde{f}_{\mu,k} &= \langle \widetilde{f}(t), \phi_{\mu,k}(t) \rangle = \langle f(t), \widetilde{\phi}_{\mu,k}(t) \rangle \,, \\ \widetilde{d}_{\mu,k} &= \langle \widetilde{f}(t), \psi_{\mu,k}(t) \rangle = \langle f(t) \widetilde{\psi}_{\mu,k}(t) \rangle \,, \end{aligned} \tag{3.3.39}$$

wobei $\widetilde{\phi}_{\mu,k}(t) = \sum_l \phi_{\mu,k}(2^\mu(t + lT) - k)$ und analog $\widetilde{\psi}(t)$ die periodischen Fortsetzungen der Skalierungs– und Waveletfunktionen sind. Weiterhin lässt sich zeigen, dass dann der *Pyramidenalgorithmus* die Form annimmt

$$\widetilde{f}_{\mu,k} = \sum_l g_{l-2k} \widetilde{f}_{\mu+1,l} = \sum_l \widetilde{g}_{l-2k} f_{\mu+1,l} \,, \tag{3.3.40}$$

$$\widetilde{d}_{\mu,k} = \sum_l h_{l-2k} \widetilde{f}_{\mu+1,l} = \sum_l \widetilde{h}_{l-2k} f_{\mu+1,l} \,. \tag{3.3.41}$$

Definition 3.9 *Eine* **diskrete Wavelet–Transformation** (DWT) *einer endlichen diskreten Funktion* $\boldsymbol{f} = [f_j]$ *ist definiert durch eine Periode der Koeffizienten in (3.3.40) und (3.3.41).*

Die inverse DWT ergibt sich durch analoge Anwendung der Synthesegleichung (3.3.33).

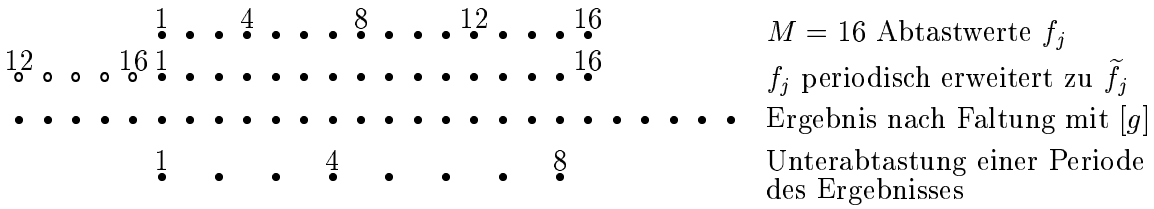


Bild 3.3.5: Die Schritte periodische Fortsetzung, Faltung, Unterabtastung bei der Wavelet Transformation, am Beispiel $f_j,\ j = 1, \ldots, 16,\ g_j,\ j = 1, \ldots, 6$

Mit (3.3.40), (3.3.41) ist die Grundlage für die diskrete Wavelet–Transformation einer endlichen Folge von Abtastwerten gegeben. Die Koeffizienten h_i, g_i hängen von der verwendeten Basis ab; für die HAAR-Basis wurden sie in (3.3.24) und (3.3.25) angegeben. Aus Bild 3.3.4 geht hervor, dass die Zahl der Koeffizienten von $f_{\mu,j}$ und $d_{\mu,j}$ in der nächst kleineren Auflösung jeweils halb so groß ist wie die der Koeffizienten von $f_{\mu+1,j}$.

Auf der Basis der periodischen Fortsetzung von f in (3.3.40), erste Teilgleichung, wird noch kurz ein Algorithmus skizziert. Die zu transformierende Funktion $[f] = [f]_{\mu+1}$ habe $M = 2^m$ Abtastwerte, die Skalierungsfunktion $[g]$ habe L solcher Werte, s. Bild 3.3.5 für $M = 16,\ L = 6$. Zunächst wird $[f]$ periodisch fortgesetzt, indem die $L-1$ letzten Abtastwerte von $[f]$ links angesetzt werden; das Ergebnis $[\widetilde{f}]$ hat nun $M + L - 1$ Abtastwerte. Aus dem Pyramidenalgorithmus (3.3.31) geht hervor, dass die Summe über l einer *Faltung* von $[f]$ mit $[g_{-l}]$ entspricht, d. h. die Werte von $[g]$ werden von rechts nach links beginnend angeordnet. Die Koeffizienten von $[h]$ werden aus (3.3.23) berechnet. Es wird $[f] = [f]_{\mu+1}$ mit $[g_{-l}]$ und mit $[h]$ gefaltet; das Ergebnis hat jeweils $M + 2 \times (L-1)$ Abtastwerte. Die Koeffizienten $[f]_\mu$ bzw. $[d]_\mu$ erhält man, wieder gemäss (3.3.31) bzw. Bild 3.3.2, durch *Unterabtastung* der Ergebnisse der Faltungen mit $[g_{-l}]$ bzw. mit $[h]$ über eine Periode, d. h. durch Auswahl von $M/2$ Koeffizienten aus den Ergebnissen dieser beiden Faltungen, beginnend ab Abtastwert L. Dieser Prozess kann dann auf den $M/2$ Koeffizienten von $[f]_\mu$ wiederholt werden. Als Beispiel werden unten in Tabelle 3.1 die Filterkoeffizienten g_l einiger Skalierungsfunktionen angegeben; die entsprechenden Waveletkoeffizienten h_l ergeben sich aus (3.3.23).

Als *Merkmale* für die Klassifikation von Mustern werden z. T. einige der Koeffizienten einer Wavelet–Transformation direkt verwendet, z. T. aber daraus Merkmale berechnet, die z. B. bestimmte Invarianzeigenschaften haben.

3.3.5 Zweidimensionale Wavelet Transformation

Mit zweidimensionalen Skalierungsfunktionen der Form (3.3.16) erhält man einen Mittelwert– oder Tiefpassanteil $f_{\mu,j,k}$ und mit zweidimensionalen Wavelets der Form in (3.3.17) erhält man nun *drei* Differenz– oder Hochpassanteile $d_{0,\mu,j,k},\ d_{1,\mu,j,k},\ d_{2,\mu,j,k}$. Diese werden berechnet, indem man eine eindimensionale Wavelet Transformation der Reihe nach auf alle *Zeilen* eines Bildes $\boldsymbol{f} = [f_{j,k}]$ anwendet und damit je Zeile einen Mittelwert- und einen Differenzanteil berechnet, wie es in Bild 3.3.4 angedeutet ist. Danach wendet man eine Wavelet Transformation der Reihe nach auf alle *Spalten* des resultierenden Koeffizientenfeldes an und erhält so vier Koeffizientenfelder. Dieser Prozess kann dann wieder auf das Feld mit den Tiefpasskoeffizienten erneut angewendet werden, wie es Bild 3.3.6 für zwei Schritte zeigt. Natürlich kann man auch erst die Spalten und dann die Zeilen transformieren. Ein Beispiel für die dreimalige diskrete

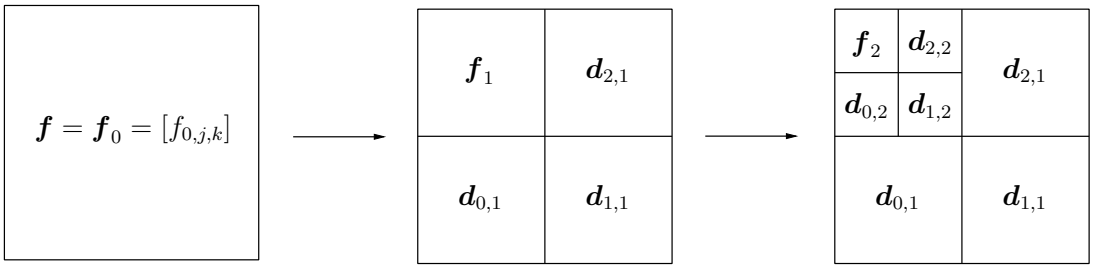


Bild 3.3.6: Zweidimensionale Wavelet Transformation als Kombination einer eindimensionalen Transformation je Zeile gefolgt von einer eindimensionalen Transformation je Spalte; das Feld $f_{\mu,j,k}$ kann anschließend erneut transformiert werden

Original transformiert mit HAAR Basis

Bild 3.3.7: Beispiel für die Wavelet Transformation eines Bildes

Wavelet–Transformation eines zweidimensionalen Musters zeigt Bild 3.3.7.

HAAR	0.707107	0.707107						
DAUBECHIES 4	0.482963	0.836516	0.224144	-0.129410				
DAUBECHIES 6	0.015656	-0.072733	-0.384865	0.852572	-0.337898	-0.0727233		
JOHNSTON 8	-0.015274	-0.099917	-0.098186	0.692937	0.692937	0.098186	0.099917	0.015274
VILLASENOR	-0.088388	-0.88388	0.707107	-0.707107	0.088388	0.088388		

Tabelle 3.1: Filterkoeffizienten g_l einiger Skalierungsfunktionen

3.4 Filterbänke (VA.2.1.3, 09.03.2004)

Eine *Filterbank* ist in der Regel eine Menge von Filtern gleichen Typs, z. B. von *Bandpässen*, mit geeignet gestuften Parametern, z. B. abgestuften Werten für die Mittenfrequenz und die Bandbreite. Die Eingangsgröße ist ein Muster, die Ausgangsgrößen der Filter werden als Merkmale verwendet. Es sind geeignete Filtertypen und Parameter zu wählen. Die Filterantwort wird oft an unterschiedlichen Orten im Bild berechnet. Orte, an denen die Filterantwort berechnet wird, sind z. B. ein regelmäßiges Gitter oder markante Bildpunkte.

3.4.1 GABOR–Filter

GABOR–Funktion

Die GABOR-*Filter* beruhen auf den GABOR-Funktionen, die eine Zerlegung in der Zeit (bzw. im Ort) und in der Frequenz ergeben, ähnlich wie die Waveletanalyse. Sie sind optimal in Ort und Frequenz konzentriert im Sinne der Unschärferelation, Satz 2.2, S. 67. Die GAUSS-Funktion ist die einzige Funktion $\mathbb{R} \to \mathbb{R}$, die das Minimum der Unschärferelation erreicht, und die eindimensionale bzw. zweidimensionale GABOR-Funktion ist die einzige Funktion $\mathbb{R} \to \mathsf{C}$ bzw. $\mathbb{R}^2 \to \mathsf{C}$, die dieses Minimum erreicht. Weiterhin haben sie eine ähnliche Charakteristik wie Zellen des visuellen Cortex. Es sind i. Allg. harmonische Funktionen, die durch eine Normalverteilung gedämpft werden.

Definition 3.10 *Die zweidimensionale* GABOR-Funktion*, auch als* GABOR-*Elementarfunktion bezeichnet, ist gegeben durch*

$$\begin{aligned} g(x,y) &= \frac{1}{2\pi\lambda\sigma^2}\exp\left[-\frac{1}{2}\left(\frac{x'^2}{\lambda^2\sigma^2}+\frac{y'^2}{\sigma^2}\right)\right]\exp[\mathrm{i}\,2\pi(\xi_0 x+\eta_0 y)]\ , \\ (x',y') &= (x\cos\phi+y\sin\phi,\ -x\sin\phi+y\cos\phi) \end{aligned} \tag{3.4.1}$$

mit dem Frequenzgang

$$\begin{aligned} G(\xi,\eta) &= \exp\left[-2\pi^2\sigma^2\left(\lambda^2(\xi-\xi_0)'^2+(\eta-\eta_0)'^2\right)\right]\ , \\ (\xi-\xi_0)' &= (\xi-\xi_0)\cos\phi+(\eta-\eta_0)\sin\phi\ , \\ (\eta-\eta_0)' &= -(\xi-\xi_0)\sin\phi+(\eta-\eta_0)\cos\phi\ . \end{aligned} \tag{3.4.2}$$

Von der aus Real– und Imaginärteil bestehenden Funktion $g(x,y)$ wird teilweise, insbesondere für Zwecke der Merkmalsgewinnung, nur der Realteil verwendet.

Es handelt sich also um eine zweidimensionale harmonische Schwingung mit Real– und Imaginärteil, die durch eine GAUSS-Funktion gedämpft wird. Die Parameter sind das Verhältnis λ der Ausdehnung der GAUSS-Funktion in x– und y–Richtung, der Winkel ϕ der Drehung der Hauptachsen der GAUSS-Funktion sowie der Winkel $\theta = \arctan(\eta_0/\xi_0)$ der zweidimensionalen harmonischen Schwingung. Die Anteile sowie das Produkt der Anteile zeigt Bild 3.4.1. Der Frequenzgang $G(\xi,\eta)$ ist ein *Bandpass* dessen Hauptachsen ebenfalls um ϕ gegen die ξ–Achse rotiert sind, mit Mittenfrequenz $(\xi_0,\ \eta_0)$, d. h. Betrag der Mittenfrequenz $\varphi_0 = \sqrt{\xi_0^2+\eta_0^2}$, und Orientierung $\theta = \arctan(\eta_0/\xi_0)$.

Die Winkel ϕ und θ können im Prinzip unabhängig voneinander gewählt werden. Bild 3.4.2 zeigt die GABOR-Funktionen für drei Werte von θ und $\phi = 0$. Die Wahl der Parameter λ, ϕ, θ

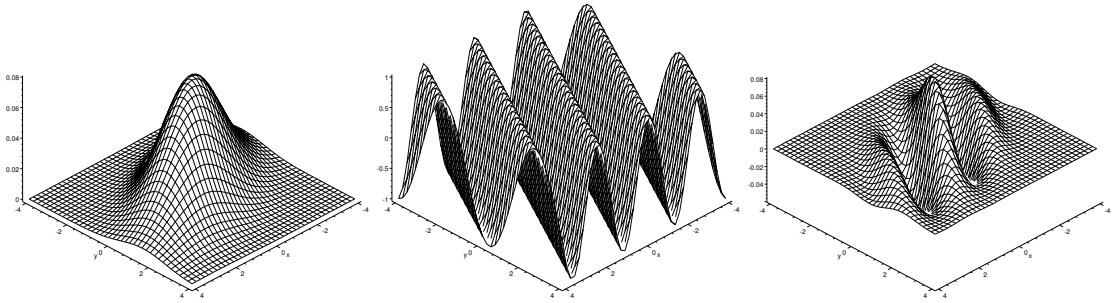


Bild 3.4.1: Eine zweidimensionale GAUSS-Funktion mit $\lambda = 2$ und $\phi = 30^o$ (links), der Realteil einer zweidimensionalen harmonischen Schwingung mit $\theta = -30^o$ (mitte), sowie das Produkt der beiden gemäß (3.4.1) (rechts)

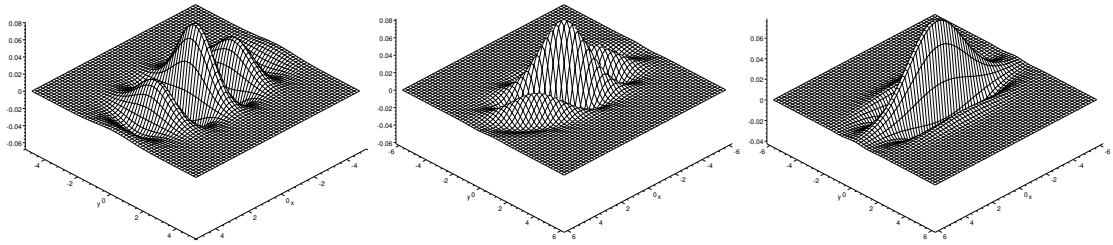


Bild 3.4.2: GABOR-Funktionen für $\phi = 0$ und $\theta = 0^o,\ 45^o,\ 90^o$ (von links nach rechts)

eröffnet Freiheitsgrade, die für die Zwecke der Merkmalsgewinnung in der Regel nur schwer nutzbar sind. Daher wird oft eine von zwei Spezialisierungen der obigen Definition 3.10 verwendet. Setzt man $\lambda = 1$, d. h. verwendet eine radialsymmetrische GAUSS-Funktion, so erübrigt sich die Angabe des Winkels ϕ. Die resultierende Form von (3.4.1) ist offensichtlich. Hält man λ variabel, so wird oft $\theta = \phi$ gesetzt (das „Gänseblümchen“). In diesem Falle reduzieren sich (3.4.1) und (3.4.2) auf

$$g(x,y) = \frac{1}{2\pi\lambda\sigma^2}\exp\left[-\frac{1}{2}\left(\frac{x'^2}{\lambda^2\sigma^2}+\frac{y'^2}{\sigma^2}\right)\right]\exp[\mathrm{i}\,2\pi\varphi_0 x'] \ , \tag{3.4.3}$$

$$G(\xi,\eta) = \exp\left[-2\pi^2\sigma^2\left(\lambda^2(\xi'-\varphi_0)^2+\eta'^2\right)\right] \ , \tag{3.4.4}$$

mit (x', y') und (ξ', η') wie oben. Diese Funktionen $g(x, y)$ zeigt Bild 3.4.3 oben und unten ihren Frequenzgang $G(\xi, \eta)$.

GABOR–Filter

Bei einem GABOR-*Filter* wird eine GABOR-Funktion gemäß (3.4.1) oder (3.4.3) als Gewichtsfunktion verwendet, die wie in (3.4.12) diskretisiert wird. Das Filter ist durch die Parameter $(\lambda,\ \sigma,\ \phi,\ \xi_0,\ \eta_0)$ bestimmt. Ihr Entwurf ist also nichttrivial, Beschränkungen, z. B. wie oben erwähnt auf $\lambda = 1$ oder $\phi = \theta$, also naheliegend. Die Faltung eines Bildes $f(x, y)$ mit (3.4.1) ergibt gemäß (2.3.13), S. 89,

$$h(x,y) = \int\int f(u,v)g(x-u,y-v)\,\mathrm{d}u\,\mathrm{d}v \ . \tag{3.4.5}$$

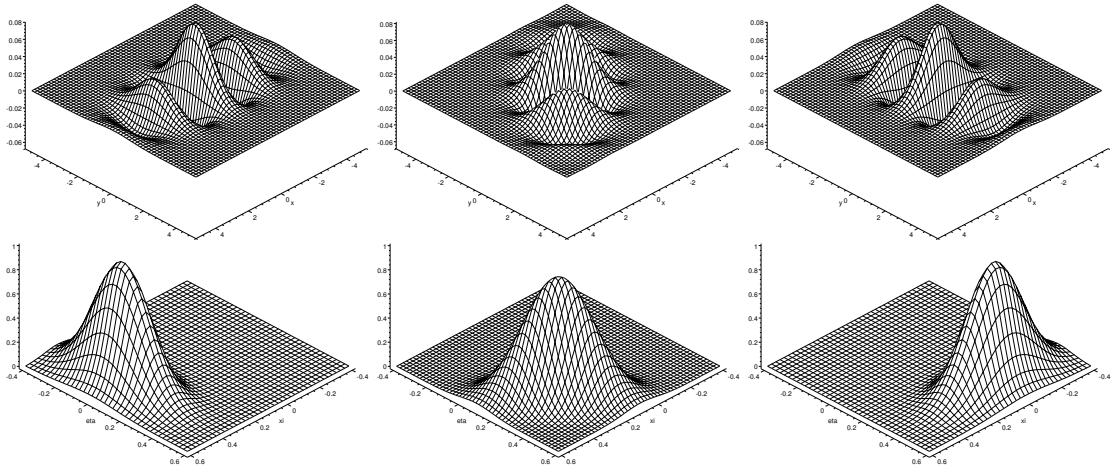


Bild 3.4.3: Das Bild zeigt *oben* von links nach rechts GABOR-Funktionen für $\phi = \theta = 0^o,\ 45^o,\ 90^o$ und *unten* von links nach rechts den Frequenzgang dieser Funktionen

Das Ergebnis der Faltung lässt sich über die GAUSS-gefensterte FOURIER-Transformation berechnen. Diese ist ein Spezialfall einer „Kurzzeittransformation“ (s. Abschnitt 3.2.3 und Abschnitt 3.6.1), hier im Orts– und nicht im Zeitbereich, und entsteht dadurch, dass man eine Fensterfunktion endlicher Ausdehnung, in diesem Falle die GAUSS-Funktion aus (3.4.1), an einer Stelle (x_0, y_0) des Bildes positioniert, beide multipliziert und das Ergebnis FOURIER-transformiert. Durch die Fensterung wird also ein lokaler Bildbereich herausgeschnitten und dessen Grauwerte mit der Fensterfunktion bewichtet. Man erhält mit (3.2.47), S. 176, für die gefensterte FOURIER-Transformierte von f

$$\begin{aligned} F_{[x_0,y_0]}(\xi,\eta) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\left(\frac{1}{2\pi\sigma^2}\exp\left[-\frac{1}{2}\left(\frac{(x-x_0)^2}{\sigma^2}+\frac{(y-y_0)^2}{\sigma^2}\right)\right]\right) \\ &\quad \exp[-\mathrm{i}\,2\pi(\xi x+\eta y)]\,\mathrm{d}x\,\mathrm{d}y\ . \end{aligned} \tag{3.4.6}$$

Es gilt der Satz

Satz 3.5 *Das Ergebnis der* GABOR*-Filterung mit Filterparametern* $(\sigma,\ \xi_0,\ \eta_0)$*, angewendet am Ort* (x_0, y_0)*, stimmt bis auf einen komplexen Faktor, der den Betrag 1 hat, mit dem Ergebnis der* GAUSS*-gefensterten* FOURIER*-Transformation an der Frequenz* (ξ_0, η_0) *überein*

$$\begin{aligned} h(x_0,y_0) &= \alpha F_{[x_0,y_0]}(\xi_0,\eta_0) \\ &= \exp[\mathrm{i}\,2\pi(\xi_0 x_0+\eta_0 y_0)]\,F_{[x_0,y_0]}(\xi_0,\eta_0)\ , \\ |h(x_0,y_0)| &= |F_{[x_0,y_0]}(\xi_0,\eta_0)|\ . \end{aligned} \tag{3.4.7}$$

Beweis*: s. z. B. [Dunn und Higgins, 1995], bzw. durch Umformungen des Faltungsintegrals (3.4.5).*

Dieser Satz besagt insbesondere, dass die GAUSS-gefensterte FOURIER-Transformation – mit variablen Werten für (ξ, η) – einer GABOR-Filterung mit einer *Menge* von GABOR-Filtern mit einem Kontinuum von Parameterwerten für (ξ_0, η_0) entspricht. Diese ist effizient über den Multiplikationssatz, Satz 2.12, S. 96, berechenbar. Die Auswertung wird natürlich diskret erfol-

gen, d. h. für Folgen von Abtastwerten.

Ein GABOR-Filter gemäß (3.4.3) hat eine bestimmte Bandbreite, innerhalb derer der Frequenzgang auf den halben Betrag abfällt. In Polarkoordinaten gehört zu dieser Bandbreite eine Frequenz B_r, gemessen in Oktaven (zwischen den Frequenzen f_u bis f_o liegen $\log_2(f_o/f_u)$ Oktaven), und ein Winkel B_α, gemessen im Bogen– oder Gradmaß. Diese sind

$$B_r = \log_2\left[\frac{2\pi\varphi_0\lambda\sigma + a}{2\pi\varphi_0\lambda\sigma - a}\right] , \qquad a = \sqrt{2\,\ln 2}\,, \tag{3.4.8}$$

$$B_\alpha = 2\arctan\left[\frac{a}{2\pi\varphi_0\sigma}\right] . \tag{3.4.9}$$

Die Realisierung erfolgt durch diskrete Operationen. Daher muss die kontinuierliche GABOR-Funktion *abgetastet* werden, wobei das Abtasttheorem Satz 2.1, S. 65, zu beachten ist. Aus (3.4.4) folgt, dass dieses nur näherungsweise eingehalten werden kann. Wenn man als „kritische“ Abtastrichtung die Richtung θ betrachtet und die verfälschende Energie auf 10% beschränkt, ergibt sich eine erforderliche Abtastfrequenz f_s von

$$2,6\,\varphi_0 \le f_s \le 4,6\,\varphi_0 \quad \text{für} \quad 0,5 \le B \le 3\,. \tag{3.4.10}$$

Wenn man die verfälschende Energie auf nur 1% beschränkt, ergibt sich

$$3,2\,\varphi_0 \le f_s \le 7\,\varphi_0 \quad \text{für} \quad 0,5 \le B \le 3\,, \tag{3.4.11}$$

wobei die kleinere Abtastfrequenz jeweils für die kleinere Bandbreite gilt. Der Realteil der diskreten Gewichtsfunktion ergibt sich damit aus (3.4.3) für $\phi = \theta = 0$ zu

$$g_{j,k} = \frac{1}{2\pi\lambda\sigma^2}\exp\left[-\frac{1}{2}\left(\frac{j^2}{\lambda^2\sigma^2 f_s^2} + \frac{k^2}{\sigma^2 f_s^2}\right)\right]\exp\left[\mathrm{i}\,2\pi\varphi_0\frac{j}{f_s}\right] \tag{3.4.12}$$

GABOR–Merkmale

Von den GABOR-Funktionen wurden in der Literatur durch unterschiedliche Wahl der Parameter und unterschiedliche weitere Verarbeitungsschritte eine Reihe von Merkmalen unter Bezeichnungen wie GABOR-Filter, –Koeffizienten, –Wavelets, –Merkmale oder –Entwicklung in der Literatur eingeführt.

3.4.2 GAUSS-Filter

Das GAUSS-Filter wurde bereits in (2.3.41), S. 101, zum Zwecke der Störungsreduktion vorgestellt. Die Richtungsableitung nach x ist

$$g_x(x,y) = -\frac{x}{2\pi\sigma^4}\exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] . \tag{3.4.13}$$

Teilweise wird beim GAUSS-Filter der normierende Vorfaktor $1/(2\pi\sigma^2)$ fortgelassen, sodass sich die Richtungsableitung

$$g'_x(x,y) = -\frac{x}{\sigma^2}\exp\left[-\frac{x^2+y^2}{2\sigma^2}\right]$$

ergibt. Wenn man die Richtungsableitungen nach x und y berechnet hat, lassen sich die Richtungsableitungen zu einer beliebigen Richtung α daraus berechnen

$$g_\alpha(x,y) = g_x(x,y)\,\cos\alpha + g_y(x,y)\,\sin\alpha\ . \tag{3.4.14}$$

Dieses ist ein Beispiel für ein *steuerbares Filter*, das mit wenigen Basisfunktionen – in diesem Falle zwei – die Filterantwort für beliebige Parametrierungen – in diesem Falle Winkel α – berechnet.

Eine Modifikation ist die Verwendung unterschiedlicher Werte σ_x bzw. σ_y für die Koordinaten x bzw. y. Höhere Ableitungen können ebenfalls verwendet werden. Durch Kombination von GAUSS-Filtern mit unterschiedlichen Werten von σ und deren Richtungsableitungen samt Betrag entstehen Filter, deren Ausgangswerte als Merkmale genutzt werden. Ein Beispiel ist die Verwendung von fünf Werten für σ und neun Richtungsableitungen, was zu einem Merkmalsvektor mit $n = 45$ Komponenten je Ort im Bild, an dem die Filterantworten berechnet werden, führt. Ein anderes Beispiel ist die Mittelung von Filterantworten mit unterschiedlichen Richtungsableitungen aber gleichem Wert von σ, um eine Richtungsunabhängigkeit der Merkmale zu erreichen.

3.5 Andere heuristische Verfahren (VA.1.3.2, 08.04.2004)

3.5.1 R–Transformation

Mit Satz 3.2, S. 170, war es möglich, Koeffizienten c_ν gemäß (3.2.23) zu bestimmen, die translationsinvariant sind. Eine Modifikation der WALSH–HADAMARD-Transformation (WHT) wurde unter der Bezeichnung **R–Transformation** oder RAPID Transformation angegeben. Sie ist definiert durch

$$\begin{aligned} f_{2j}^{l} &= |\, f_j^{l-1} + f_{j+\frac{M}{2l}}^{l-1} \,|\,, \quad M = 2^q\,, \quad l = 1, 2, \ldots, q \quad , \\ f_{2j+1}^{l} &= |\, f_j^{l-1} - f_{j+\frac{M}{2l}}^{l-1} \,|\,, \quad j = 0, 1, \ldots, \tfrac{M}{2l} - 1 \quad , \\ f_k^0 = f_k \quad &\text{und} \quad f_k^q = c_k\,, \qquad k = 0, 1, \ldots, M-1 \quad . \end{aligned} \tag{3.5.1}$$

Es handelt sich hier um eine nichtlineare Transformation. Der Signalflussgraph dieser Transformation ist übrigens identisch dem der schnellen WHT in (3.2.7), jedoch fehlt bei der WHT die Betragsbildung. Mit (3.5.1) ist also auch eine einheitliche Darstellung der schnellen WHT gegeben (jedoch in anderer Anordnung als in (3.2.62), (3.2.63)), wenn die Betragsbildung unterbleibt. Für die R–Transformation gilt

Satz 3.6 *Die mit der R–Transformation gemäß (3.5.1) berechneten Merkmale c_k sind* invariant *gegenüber einer zyklischen Verschiebung wie in Bild 3.2.2*

$$[f_0, f_1, \ldots, f_{M-1}] \longrightarrow [f_{0+m}, f_{1+m}, \ldots, f_{M-1+m}]\,,$$

wenn man $M + \nu = \nu$ setzt, und gegenüber einer Spiegelung

$$[f_0, f_1, \ldots, f_{M-1}] \longrightarrow [f_{M-1}, \ldots, f_1, f_0]$$

des Musters $\boldsymbol{f}$.
Beweis*: s. z. B. [Reitboeck und Brody, 1969]*

3.5.2 Momente

Momente eines Musters $f(x, y)$ wurden bereits in (2.5.31), S. 128, definiert. Hier wird von *Zentralmomenten*

$$\mu_{pq} = \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} (x_j - x_s)^p (y_k - y_s)^q f_{jk}\, \Delta x\, \Delta y \tag{3.5.2}$$

ausgegangen. Der Schwerpunkt (x_s, y_s) wurde im Zusammenhang mit (2.5.32) definiert. Die Koordinaten x_j, y_k ergeben sich aus (2.1.1); ist $x_0 = 0$ und $\Delta x = 1$, so ist $x_j = j$. Die diskrete Version (3.5.2) der in (2.5.31) eingeführten Momente eignet sich unmittelbar für die Verarbeitung von Abtastwerten, hat aber natürlich einen Verlust an Genauigkeit bei der Berechnung der Momente zur Folge. Die Zentralmomente sind translationsinvariant. Eine Menge von sieben Merkmalen $c_\nu,\ \nu = 1, \ldots, 7$, die aus Zentralmomenten bis zur Ordnung $p + q = 3$ berechnet werden und rotationsinvariant sind, ist

$$c_1 \;=\; \mu_{20} + \mu_{02}\,,$$

$$
\begin{aligned}
c_2 &= (\mu_{20}-\mu_{02})^2+4\mu_{11}^2\ ,\\
c_3 &= (\mu_{30}-3\mu_{12})^2+(3\mu_{21}-\mu_{03})^2\ ,\\
c_4 &= (\mu_{30}+\mu_{12})^2+(\mu_{21}+\mu_{03})^2\ ,\\
c_5 &= (\mu_{30}-3\mu_{12})(\mu_{30}+\mu_{12})((\mu_{30}+\mu_{12})^2-3(\mu_{12}+\mu_{03})^2)\\
&\quad +(3\mu_{21}-\mu_{03})(\mu_{12}+\mu_{03})(3(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2)\ ,\\
c_6 &= (\mu_{20}-\mu_{02})((\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2)\\
&\quad +4\mu_{11}(\mu_{30}+\mu_{12})(\mu_{21}+\mu_{03})\ ,\\
c_7 &= (3\mu_{21}-\mu_{03})(\mu_{30}+\mu_{12})((\mu_{30}+\mu_{12})^2-3(\mu_{21}+\mu_{03})^2)\\
&\quad +(3\mu_{12}-\mu_{30})(\mu_{21}+\mu_{03})(3(\mu_{30}+\mu_{12})^2-(\mu_{21}+\mu_{03})^2)\ .
\end{aligned}
\tag{3.5.3}
$$

Die Invarianz gilt für die Berechnung der Momente mit (2.5.31); bedingt durch Ungenauigkeiten der diskreten Form sind gewisse Abweichungen möglich. Eine Größeninvarianz, d. h. Invarianz gegenüber der Koordinatentransformation

$$x' = ax \quad \text{und} \quad y' = ay \tag{3.5.4}$$

wird durch Verwendung der Momente

$$\mu'_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(p+q)/2}} \tag{3.5.5}$$

zur Berechnung der c_ν erreicht, oder durch Verwendung von

$$
\begin{aligned}
c'_2 &= \frac{c_2}{r^4}\ ,\\
c'_3 &= \frac{c_3}{r^6}\ , \qquad c'_4 = \frac{c_4}{r^6}\ ,\\
c'_6 &= \frac{c_6}{r^8}\ ,\\
c'_5 &= \frac{c_5}{r^{12}}\ , \quad c'_7 = \frac{c_7}{r^{12}}\ ,
\end{aligned}
\tag{3.5.6}
$$

wobei r die in (2.5.33) eingeführte Größe $r = \sqrt{\mu_{20}+\mu_{02}}$ ist. Die Verwendung derartiger invarianter Momente als Merkmale basiert auf der Tatsache, dass unter bestimmten Voraussetzungen ein Muster eindeutig durch seine Momente $\mu_{pq},\ p,q = 0,1,2,\ldots$ gekennzeichnet wird.

Eine Verallgemeinerung der Momente sind die LEGENDRE- und ZERNIKE-Momente sowie Momente für dreidimensionale Objekte. Zur effizienten numerischen Berechnung aller erwähnten Momente wird auf die Literatur verwiesen.

Definition 3.11 *Die* LEGENDRE-**Momente** *sind definiert durch*

$$\lambda_{pq} = \frac{(2p+1)(2q+1)}{4}\int_{-1}^{+1}\int_{-1}^{+1} P_p(x)P_q(y)f(x,y)\,\mathrm{d}x\,\mathrm{d}y\ , \tag{3.5.7}$$

wobei $P_p(x)$ das Legendre Polynom der Ordnung p ist.

Die LEGENDRE-**Polynome** sind in $(-1,+1)$ rekursiv definiert durch

$$
\begin{aligned}
P_p(x) &= \frac{1}{p}\left[(2p-1)xP_{p-1}(x)-(p-1)P_{p-2}(x)\right]\ , \quad p=2,3,\ldots\ ,\\
P_0(x) &= 1\ , \quad P_1(x)=x\ .
\end{aligned}
\tag{3.5.8}
$$

Für ein Muster $f(r,\phi)$ in Polarkoordinaten gilt:

Definition 3.12 *Die* ZERNIKE-**Momente** *sind definiert durch*

$$\zeta_{pq} = \frac{p+1}{\pi} \int_0^1 \int_{-\pi}^{+\pi} R_{pq}(r) \exp[-\mathrm{i}\, q\phi]\, f(r,\phi) r \,\mathrm{d}r \,\mathrm{d}\phi \,, \tag{3.5.9}$$

$$R_{pq}(r) = \sum_{s=0}^{(p-|q|)/2} (-1)^s \frac{(p-s)!}{s! \left(\frac{p+|q|}{2} - s\right)! \left(\frac{p-|q|}{2} - s\right)!} r^{(p-2s)} \,, \tag{3.5.10}$$

$$r = \sqrt{x^2+y^2} \,, \quad \phi = \tan^{-1}\left(\frac{y}{x}\right)$$
$$-1 < x, y < +1 \,, \quad n > 0 \,, \quad 0 \le |m| \le n$$

Auch die Funktionen $R_{pq}(r)$ können rekursiv berechnet werden. Die ζ_{pq} werden oft in Real– und Imaginärteil zerlegt berechnet.

Definition 3.13 Momente *für dreidimensionale Objekte bzw. Volumina sind definiert durch*

$$m_{pqr} = \iiint x^p y^q z^r f(x,y,z) \,\mathrm{d}x \,\mathrm{d}y \,\mathrm{d}z \,. \tag{3.5.11}$$

3.5.3 Merkmalsfilter

Wie in (2.3.38) werden mit $\boldsymbol{s}$ ein ideales Muster und mit $\boldsymbol{n}$ eine Störung sowie mit $\boldsymbol{f}_0, \boldsymbol{f}_1$ zwei beobachtete Muster bezeichnet, die durch

$$\begin{aligned} \boldsymbol{f}_0 &= \boldsymbol{n} \,, \quad \text{mit} \quad E\{\boldsymbol{n}\} = \boldsymbol{0} \,, \\ \boldsymbol{f}_1 &= \boldsymbol{s} + \boldsymbol{n} \end{aligned} \tag{3.5.12}$$

definiert sind. Gesucht ist ein lineares System $[g_j]$, dessen Ausgangsgröße h_j für einen bestimmten Index $j = j_0 = \text{const}$ eine möglichst gute Unterscheidung zwischen $\boldsymbol{f}_0$ und $\boldsymbol{f}_1$ erlaubt. Mit (2.3.14) gilt für ein beobachtetes Muster $\boldsymbol{f}$

$$\begin{aligned} h_{j_0} &= \sum_{\mu=0}^{M-1} f_\mu g_{j_0-\mu} \qquad j_0 = \text{const} \,, \\ &= \bar{\boldsymbol{g}}^{\mathsf{T}} \boldsymbol{f} \,, \end{aligned} \tag{3.5.13}$$

wobei $\bar{\boldsymbol{g}}$ ein Vektor mit den Komponenten $g_{j_0}, g_{j_0-1}, \ldots, g_{j_0-M+1}$ ist. Ist $\boldsymbol{f} = \boldsymbol{f}_0$, so wird die Energie der Ausgangsgröße für $j = j_0$ als *Rauschenergie* bezeichnet und mit

$$P_n = E\{(\bar{\boldsymbol{g}}^{\mathsf{T}}\boldsymbol{n})(\bar{\boldsymbol{g}}^{\mathsf{T}}\boldsymbol{n})^{\mathsf{T}}\} = \bar{\boldsymbol{g}}^{\mathsf{T}} \boldsymbol{K}_n \bar{\boldsymbol{g}} \tag{3.5.14}$$

definiert, wobei $\boldsymbol{K}_n$ die Kovarianzmatrix der Störung $\boldsymbol{n}$ ist. Ist $\boldsymbol{f} = \boldsymbol{s}$, so wird die *Signalenergie* mit

$$P_s = (\bar{\boldsymbol{g}}^{\mathsf{T}} \boldsymbol{s})^2 \tag{3.5.15}$$

definiert. Gesucht ist der Vektor $\bar{\boldsymbol{g}}$, für den das *Signal-zu-Rausch-Verhältnis*

$$\mathrm{SNR} = \frac{P_s}{P_n} = \frac{(\bar{\boldsymbol{g}}_t \boldsymbol{s})^2}{\bar{\boldsymbol{g}}_t \boldsymbol{K}_n \bar{\boldsymbol{g}}} \tag{3.5.16}$$

maximiert wird.

Satz 3.7 *Der Vektor $\bar{\boldsymbol{g}}$, der (3.5.16) maximiert, ist gegeben durch*

$$\bar{\boldsymbol{g}} = \frac{\bar{\boldsymbol{g}}^{\mathsf{T}}\boldsymbol{K}_n\bar{\boldsymbol{g}}}{\bar{\boldsymbol{g}}^{\mathsf{T}}\boldsymbol{s}}\boldsymbol{K}_n^{-1}\boldsymbol{s} = \alpha\boldsymbol{K}_n^{-1}\boldsymbol{s}\ . \tag{3.5.17}$$

Dabei ist α eine reelle Zahl, die man beispielsweise auf $\alpha = 1$ normieren kann, da (3.5.16) unabhängig vom Wert von α ist. Dann ergibt sich

$$\bar{\boldsymbol{g}} = \boldsymbol{K}_n^{-1}\boldsymbol{s}\ . \tag{3.5.18}$$

Beweis: Zunächst wird daran erinnert, dass für einen Vektor $\boldsymbol{x}$ und eine symmetrische Matrix $\boldsymbol{A}$ die Beziehung

$$\frac{\partial(\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x})}{\partial\boldsymbol{x}} = 2\boldsymbol{A}\boldsymbol{x} \tag{3.5.19}$$

gilt. Leitet man (3.5.16) nach $\bar{\boldsymbol{g}}$ ab und setzt das Ergebnis Null, so erhält man

$$\begin{aligned} 0 &= \frac{\partial(P_s/P_n)}{\partial\bar{\boldsymbol{g}}} \\ &= (\bar{\boldsymbol{g}}^{\mathsf{T}}\boldsymbol{K}_n\bar{\boldsymbol{g}})\boldsymbol{s}(\boldsymbol{s}^{\mathsf{T}}\bar{\boldsymbol{g}}) - (\bar{\boldsymbol{g}}^{\mathsf{T}}\boldsymbol{s})(\boldsymbol{s}^{\mathsf{T}}\bar{\boldsymbol{g}})\boldsymbol{K}_n\bar{\boldsymbol{g}}\ . \end{aligned} \tag{3.5.20}$$

Beachtet man noch, dass $(\boldsymbol{s}^{\mathsf{T}}\bar{\boldsymbol{g}})$ ein Skalar ist, so folgt daraus sofort (3.5.17). Damit ist gezeigt, dass Satz 3.7 eine notwendige Bedingung ist.

Man bezeichnet das durch $\bar{\boldsymbol{g}}$ definierte lineare System auch als **angepasstes Filter** ("matched filter"). Ist die Störung ein weißes Rauschen, so ist $\boldsymbol{K}_n = \boldsymbol{I}$ und $\bar{\boldsymbol{g}} = \boldsymbol{s}$. Eine Verallgemeinerung auf den mehrdimensionalen Fall ist ohne weiteres möglich; man braucht nur $\bar{\boldsymbol{g}}$ und $\boldsymbol{s}$ als mehrdimensionale Folge aufzufassen. Die Formulierung ist analog im kontinuierlichen Fall möglich. Ohne auf Einzelheiten einzugehen, wird erwähnt, dass sich aus dem Multiplikationssatz Satz 2.12, S. 96, für den Frequenzgang des angepassten Filters im kontinuierlichen Fall

$$G(\xi,\eta) = \frac{S^*(\xi,\eta)}{|N(\xi,\eta)|^2} \tag{3.5.21}$$

ergibt, wobei G, S, N die FOURIER-Transformierten von g, s, n sind und S^* die konjugiert komplexe von S ist. Als *phasenangepasstes Filter* wird

$$G_{\mathrm{p}}(\xi,\eta) = \frac{S^*(\xi,\eta)}{|S^*(\xi,\eta)|} \tag{3.5.22}$$

verwendet.

Die Operation (3.5.13) lässt sich als *Korrelation* zwischen dem Vektor $\bar{\boldsymbol{g}}$ und dem beobachteten Muster $\boldsymbol{f} = \boldsymbol{f}_0$ oder $\boldsymbol{f} = \boldsymbol{f}_1$ auffassen. Das Ausgangssignal h_{j_0} an der Stelle $j = j_0$ weist dann eine Korrelationsspitze auf, wenn in $\boldsymbol{f}$ das ideale Signal $\boldsymbol{s}$ enthalten ist, da dann (3.5.16) maximiert wird. Da man i. Allg. die Stelle $j = j_0$ *nicht* im Voraus kennt, ist (3.5.13) für verschiedene Werte von j auszuwerten. Wenn h_j für einen Wert von j über einer Schwelle liegt, so wird angenommen, dass $\boldsymbol{f} = \boldsymbol{f}_1$ ist. Das Signal $\boldsymbol{s}$ wurde hier als deterministisch angenommen, jedoch ist auch eine Formulierung für ein stochastisches Signal $\boldsymbol{s}$ mit Mittelwert $\boldsymbol{m}_s$ und Kovarianzmatrix $\boldsymbol{K}_s$ möglich, auf die hier aber verzichtet wird.

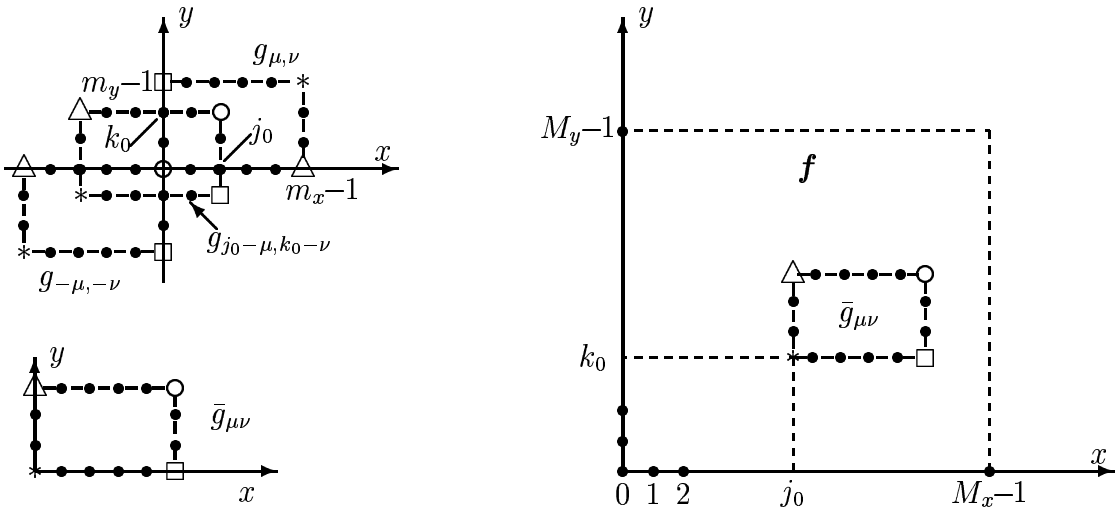


Bild 3.5.1: Zur Berechnung der Antwort eines angepassten Filters im zweidimensionalen Fall

Die Ausgangsgleichung (3.5.12) lässt verschiedene Interpretationen zu. Im Kontext der Merkmalsgewinnung wird man $\boldsymbol{f}$ als ein Muster auffassen, das daraufhin zu untersuchen ist, ob es ein bestimmtes Merkmal $\boldsymbol{s}$ – z. B. eine Linienkreuzung, ein gerades Linienelement oder irgendein anderes einfacheres Bestandteil – enthält oder nicht; die Technik wird auch für komplexere Einheiten wie das Auge oder ganze Objekte wie das Gesicht eingesetzt. Der Ort, an dem das Merkmal zu erwarten ist, ist dabei unbekannt. Die Suche nach dem Merkmal kann dann durch die Suche nach einer genügend großen Korrelationsspitze realisiert werden. Man bezeichnet das jeweilige Filter auch als **Merkmalsfilter**. Die beschriebene Vorgehensweise hat einige Nachteile. Die Korrelation (3.5.13) und insbesondere die Suche nach der Korrelationsspitze erfordert erheblichen Rechenaufwand, der jedoch durch Rückgriff auf (2.1.14), S. 64, reduziert werden kann. Das Modell, auf dem (3.5.12) beruht, ist nur eingeschränkt brauchbar, da man als Störung $\boldsymbol{n}$ das gesamte Muster, ausgenommen das Merkmal, auffasst und diese zudem zur Vereinfachung meist als weißes Rauschen betrachtet. Noch gravierender ist, dass jede Schwankung in der Form des Merkmals gegenüber der in $\bar{\boldsymbol{g}}$ angenommenen Form sich auf die Korrelationsspitze auswirkt. Aus diesen Gründen ist die Anwendbarkeit von (3.5.13) im Einzelfall genau zu prüfen.

Die Anwendung dieser Technik auf zweidimensionale Folgen zeigt Bild 3.5.1. Zunächst entnimmt man dem Bild eine schematisierte Darstellung der Gewichtsfunktion $[g_{\mu\nu}],\ \mu = 0,1,\ldots,m_x-1,\ \nu = 0,1,\ldots,m_y-1$ des linearen Systems sowie $[g_{j_0-\mu,k_0-\nu}]$. Für die Auswertung von (3.5.13) ist es offensichtlich zweckmäßig, eine Folge

$$\bar{\boldsymbol{g}} = [\bar{g}_{\mu\nu}] = [g_{m_x-1-\mu,m_y-1-\nu}] \tag{3.5.23}$$

zu definieren und neue Werte j_0, k_0 wie in Bild 3.5.1 zu wählen. Dann erhält man

$$h_{j_0,k_0} = \sum_{\mu=0}^{m_x-1} \sum_{\nu=0}^{m_y-1} f_{j_0+\mu,k_0+\nu}\, \bar{g}_{\mu\nu}\ . \tag{3.5.24}$$

Die Folge $\bar{\boldsymbol{g}}$, die auch als *Maske* oder *Schablone* (“template”) bezeichnet wird, ergibt sich aus Satz 3.7. Ist beispielsweise $\boldsymbol{f}$ das Bild eines integrierten Schaltkreises und $\boldsymbol{s}$ eine Teilstruktur,

so kann man erwarten, dass diese Teilstruktur nur geringfügigen fertigungsbedingten Toleranzen und Fehlern unterworfen ist. In diesem Fall ist es möglich, die Teilstruktur mit (3.5.24) zu suchen. Dagegen sind bei handgedruckten Schriftzeichen erhebliche Schwankungen in Formeigenschaften zu erwarten, sodass diese Vorgehensweise problematischer ist. Die Indizes j_0, k_0 können auch so festgelegt werden, dass sie nicht wie in (3.5.24) und Bild 3.5.1 am linken unteren Rand von $\bar{\boldsymbol{g}}$ liegen, sondern beispielsweise in der Mitte. Die offensichtliche Modifikation von (3.5.24) wird nicht extra angegeben.

Statt des ursprünglich gegebenen Bildes wird oft ein bandpassgefiltertes verwendet, um Störungen zu reduzieren und Kanten hervorzuheben; eine Normierung auf Mittelwert Null und Streuung Eins mit (2.5.47), S. 131, ist zweckmäßig. Zur Reduktion des Aufwandes kann eine Auflösungshierarchie verwendet werden. Filterung, Normierung und Auflösungshierarchie sind natürlich sowohl auf das Bild als auch die Schablone anzuwenden. Die Zahl der Fehldetektionen kann reduziert werden, indem sowohl Filter für das Merkmal oder Objekt als auch solche für „Nicht-Objekte“ bereitgestellt werden.

3.5.4 Kennzahlen

Die obigen Verfahren basieren auf bekannten und gegebenenfalls modifizierten Verfahren, die eine mathematische Grundlage haben. Die Heuristik liegt darin, diese Verfahren für die Merkmalsgewinnung heranzuziehen, obwohl sie dafür ursprünglich nicht entwickelt wurden – man vergleiche Postulat 2 und 3 in Abschnitt 1.3. Daneben gibt es weitere heuristische Verfahren zur Merkmalsgewinnung, die hier unter der Bezeichnung „Kennzahlen“ zusammengefasst werden. Es handelt sich um Messwerte, Rechengrößen und Parameter, die weitgehend intuitiv und experimentell festgelegt werden. Ohne Anspruch auf Vollständigkeit werden dafür einige Beispiele gegeben.

Durch Schnittpunkte mit geeignet gewählten Testlinien lassen sich eine Reihe von Merkmalen gewinnen, die für eine Klassifikation oder zumindest für die Auswahl einiger weniger möglicher Klassen ausreichen. Bild 3.5.2a zeigt zwei Beispiele dafür. Wenn das Objekt sich in definierter Winkellage in einem Intervall $x_0 \leq x \leq x_1,\ y_0 \leq y \leq y_1$ befindet, sind horizontale und vertikale Testlinien geeignet. Als Kennzahlen oder Merkmale verwendet man beispielsweise die Zahl der Schnittpunkte des Objekts mit der Linie, die Länge des im Objekt liegenden Teils der Linie oder die Koordinaten der Schnittpunkte. Wenn die Winkellage nicht bekannt ist, kann man den Ursprung eines Polarkoordinatensystems in den Schwerpunkt des Objekts legen. Als Testlinien verwendet man Radien in konstantem Winkelabstand. Neben den oben erwähnten Merkmalen eignet sich zur Charakterisierung des Objektumrisses insbesondere der Abstand zwischen Koordinatenursprung und dem am weitesten entfernten Schnittpunkt zwischen Objekt und Testlinie. Trägt man diese Abstände über dem Winkel auf, so verursacht eine Rotation des Objekts eine Translation dieser Kurve. Ein Vergleich mit Referenzobjekten (Klassen) kann beispielsweise auch mit der im vorigen Abschnitt beschriebenen Korrelationsoperation (3.5.13) erfolgen. Aus dem Index j_0 ergibt sich dann die Drehlage des Objekts.

Aus der *Projektion* des Musters auf bestimmte Geraden – vielfach werden hier die beiden Koordinatenachsen eines rechtwinkligen Systems gewählt – ergeben sich ebenfalls Kennzahlen für das Muster. Neben Merkmalen wie Zahl und Lage der Maxima und Minima kann man wiederum die Projektionskurve direkt mit Referenzkurven vergleichen. Ein Beispiel zeigt Bild 3.5.2b. Für ein Muster $f(x, y)$ ist die Projektion auf die x–Achse definiert durch

$$f(x) = \int_{-\infty}^{\infty} f(x, y)\, \mathrm{d}y\ , \tag{3.5.25}$$

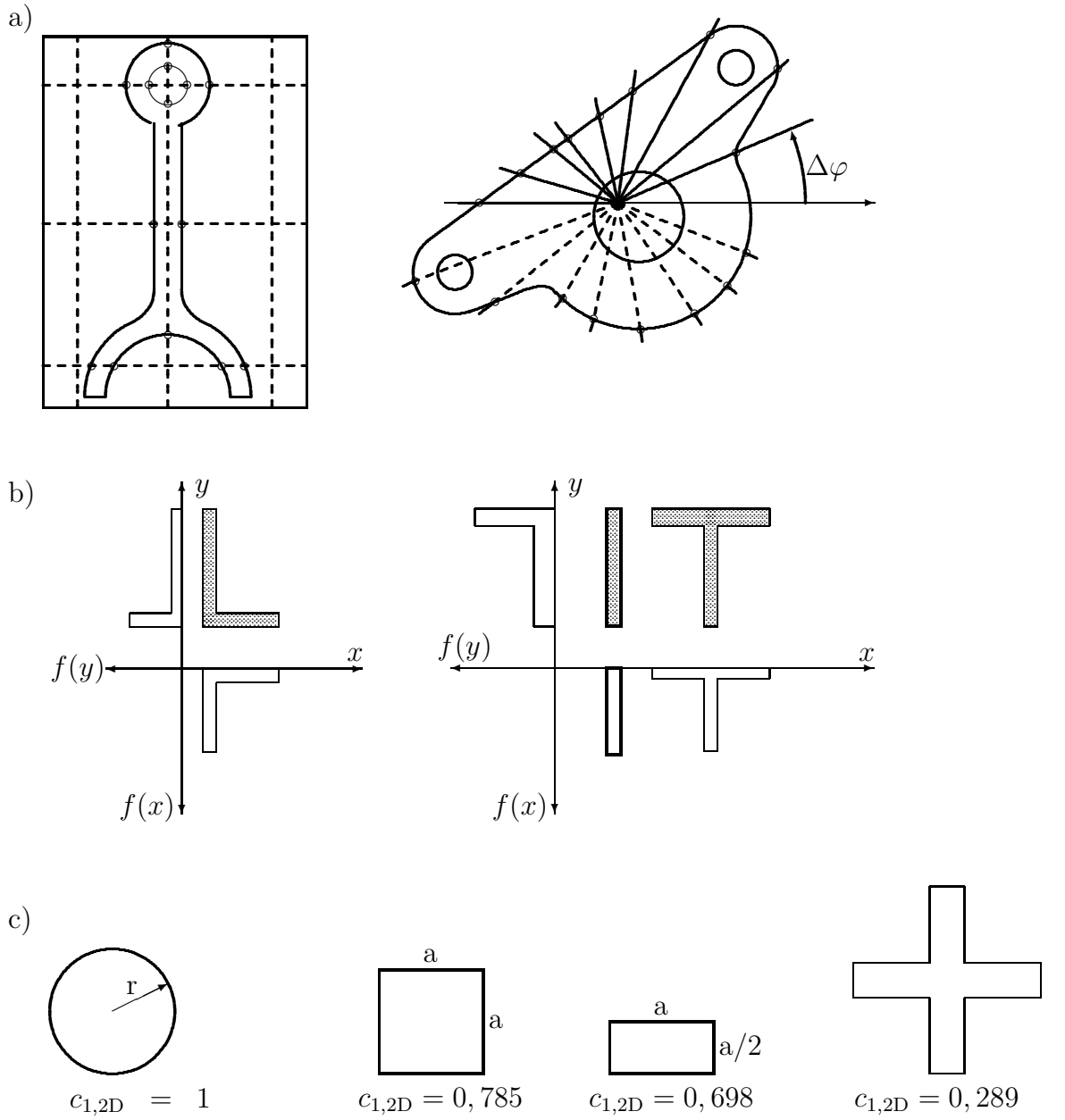


Bild 3.5.2: a) Verwendung von Testlinien, b) Projektion der Muster auf die Koordinatenachsen, c) Formfaktor

und im diskreten Falle für eine Bildmatrix $[f_{jk}]$ gilt

$$f_j = \sum_{k=0}^{M-1} f_{jk} \ , \qquad j = 0, 1, \ldots, M-1 \quad . \tag{3.5.26}$$

Entsprechendes gilt für die Projektion auf die y–Achse.

Schließlich werden auch globale Parameter oder **Formfaktoren** zur Beurteilung der *Kompaktheit* (oder Rundheit) verwendet. Für zwei– bzw. dreidimensionale Objekte werden z. B. die

Maße $c_{1,2\mathrm{D}}$ bzw. $c_{1,3\mathrm{D}}$ definiert mit

$$c_{1,2\mathrm{D}} = 4\pi\frac{A}{L^2} \in [0,1]\ , \tag{3.5.27}$$

$$c_{1,3\mathrm{D}} = 6\sqrt{\pi}\frac{V}{\sqrt{A^3}} \in [0,1]\ . \tag{3.5.28}$$

Dabei ist in (3.5.27) L der Umfang und A die Fläche eines zweidimensionalen Objekts; in (3.5.28) ist A die Oberfläche und V das Volumen eines dreidimensionalen Objekts. Jeweils für den Kreis bzw. die Kugel nehmen diese Formfaktoren ihren Maximalwert Eins an. Beispiele zeigt Bild 3.5.2c. Maße zur Beurteilung der Ähnlichkeit eines zweidimensionalen Objekts O_2 mit einem gefüllten Kreis K_2 bzw. eines dreidimensionalen Objekts O_3 mit einer gefüllten Kugel K_3 sind

$$c_{2,2\mathrm{D}} = \max\left\{1 - \frac{A(K_2 \cap \overline{O}_2) + A(\overline{K}_2 \cap O_2)}{A(O_2)},\ 0\right\}\ , \tag{3.5.29}$$

$$c_{2,3\mathrm{D}} = \max\left\{1 - \frac{V(K_3 \cap \overline{O}_3) + V(\overline{K}_3 \cap O_3)}{V(O_3)},\ 0\right\}\ . \tag{3.5.30}$$

Dabei bezeichnen $A(.)$ bzw. $V(.)$ die Fläche bzw. das Volumen des angegebenen Objekts. Kreis bzw. Kugel haben gleiche Fläche bzw. Volumen und gleichen Schwerpunkt wie die Objekte O_2 bzw. O_3.

Auch wenn die in diesem Abschnitt erwähnten Merkmale nicht zu einer genügend zuverlässigen Entscheidung für genau eine Klasse ausreichen, sind sie oft hinreichend, um eine rasche Vorauswahl von Klassen zu treffen, die dann mit zusätzlicher Information weiter bearbeitet werden.

Kennzahlen werden auch bei der Verarbeitung von Zeitfunktionen verwendet. Man gewinnt sie entweder direkt aus der Zeitfunktion $f(t)$ bzw. deren Abtastwerten f_j oder aber aus dem Spektrum von $f(t)$. Beispiele für den ersteren Fall sind die Häufigkeit von Nulldurchgängen, die Zeitabstände und Funktionsdifferenzen von relativen Extremwerten sowie Parameter von statistischen Eigenschaften der Funktionswerte wie Streuung, Schiefe oder Verteilungsdichte. Im zweiten Falle, also bei Verwendung des Spektrums, kommen abgesehen von der Häufigkeit der Nulldurchgänge im Prinzip die gleichen Kennzahlen zur Anwendung. Insbesondere bei Sprache wird oft das *Modellspektrum* (s. Abschnitt 3.6.2) verwendet, da es gegenüber dem DFT–Spektrum stark geglättet ist. Die relativen Extrema des Modellspektrums werden als *Formanten* bezeichnet und sind wichtig für die Unterscheidung von Vokalen. Dazu kommen Verhältnisse der Signalenergie in je zwei verschiedenen Frequenzbereichen und die Bestimmung der Sprachgrundfrequenz.

3.6 Merkmale für die Spracherkennung (VA.1.2.2, 06.02.2004)

3.6.1 Kurzzeittransformationen

Da ein Sprachsignal eine *zeitlich veränderliche* Struktur hat, ist es nicht sinnvoll, z. B. ein *ganzes Wort* oder gar einen ganzen Satz nach FOURIER zu transformieren bzw. daraus irgendwelche Merkmale zu extrahieren. Statt dessen zerlegt man das Muster in kurze (Zeit)–**Fenster** ("windows", "frames") von ca. 10ms Dauer, wie es schon für die gefensterte FOURIER-Transformation in Abschnitt 3.2.3 getan wurde. Generell sollte ein Fenster

- *klein* genug sein, um eine gute zeitliche Auflösung zu liefern,
- und *groß* genug sein, um eine gute Frequenzauflösung zu liefern.

Da beides wegen Satz 2.2, S. 67 oder Satz 2.3, S. 67 nicht beliebig genau möglich ist, muss ein Kompromiss geschlossen werden, indem man z. B. geeignete **Kurzzeittransformationen** wie die Kurzzeit–FOURIER-Transformation (bzw. die gefensterte FOURIER-Transformation, s. Abschnitt 3.2.3) oder andere von Zeit und Frequenz abhängige Darstellungen wie die WIGNER–VILLE Transformation definiert. Auch die bereits in Abschnitt 3.3 eingeführte Wavelet Transformation kommt in Frage.

Definition 3.14 *Die* **Kurzzeit–FOURIER-Transformation** *ist (s. auch (3.2.47), S. 176)*

$$\mathrm{KFT}(\tau,\omega) = \int f(t) w(t-\tau) \exp[-2\pi \mathrm{i}\,\omega t]\ \mathrm{d}t\ . \tag{3.6.1}$$

Dabei ist $w(t)$ eine in der Regel reelle Fensterfunktion. Beispiele für diskrete Fensterfunktionen w_ν wurden in (2.5.43) gegeben.

Die WIGNER–VILLE **Transformation** *ist*

$$\mathrm{WVT}(\tau,\omega) = \int f\left(\tau+\frac{t}{2}\right) f^*\left(\tau-\frac{t}{2}\right) \exp[-2\pi \mathrm{i}\,\omega t]\ \mathrm{d}t\ . \tag{3.6.2}$$

Im Prinzip wird mit (3.6.1) ein Muster in einzelne Zeitfenster zerlegt und je Fenster ein Merkmalsvektor berechnet. Diese Rechnung wird für eine Folge von Zeitfenstern, die das Muster überdecken, wiederholt. Es wird noch erwähnt, dass man bei Bildern ganz analog vorgeht, wenn man ein Bild in kleinere Blöcke von Bildpunkten zerlegt und je Block einen Merkmalsvektor berechnet.

3.6.2 Lineare Vorhersage

Die Methode der **linearen Vorhersage** beruht auf dem Ansatz, einen Schätzwert $\hat{f}_n$ des n–ten Wertes einer Folge $[f_j]$ von Abtastwerten mit einer *linearen* Schätzgleichung zu berechnen. In die Schätzgleichung von der Form

$$\hat{f}_n = -\sum_{\mu=1}^{m} a_\mu\, f_{n-\mu} \tag{3.6.3}$$

gehen m Werte $f_{n-1},\ldots,f_{n-m}$ ein sowie die noch zu bestimmenden *Vorhersagekoeffizienten* (oder Prädiktorkoeffizienten) a_μ. Ist $[f_j]$ eine Folge von Abtastwerten $f(j\Delta t)$ einer Zeitfunktion $f(t)$, so kann man f_n als den gerade beobachteten Wert auffassen und $f_{n-\mu},\ \mu=1,\ldots,m$

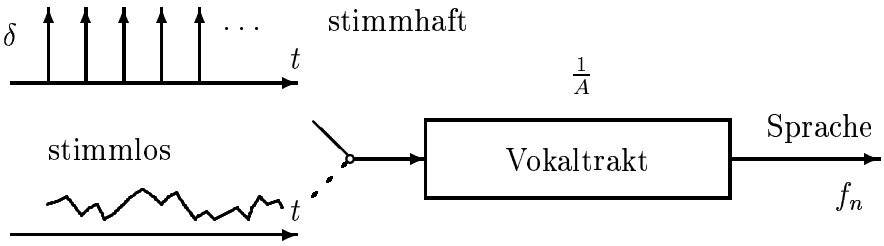


Bild 3.6.1: Die Sprachproduktion wird durch ein lineares System modelliert

sind m früher bereits beobachtete Werte, mit denen der zu erwartende Wert f_n vorhergesagt wird. Der Ansatz in (3.6.3) lässt sich im Zusammenhang mit der Spracherkennung als (lineare) Modellierung des menschlichen Stimmtraktes auffassen, wie in Bild 3.6.1 angedeutet ist, und wird auch als *autoregressives Modell* bezeichnet. Mit den Stimmbändern wird für *stimmhafte* Sprache ein periodisches Anregungssignal erzeugt, das durch eine Folge von Delta–Funktionen idealisiert wird. Das Anregungssignal wird durch den Vokaltrakt, der durch ein *lineares System* modelliert wird, gefiltert und ergibt als Ausgangsgröße das Sprachsignal. Die Impulsantwort des linearen Systems wird durch Parameter a_μ bestimmt, die sich mit der Zeit ändern und den jeweils erzeugten Laut bestimmen. Die Vorhersagekoeffizienten oder Parameter a_μ bilden die Grundlage zur Gewinnung von Merkmalen. Die Vorstellung ist also, dass für Muster einer Klasse Ω_κ Funktionswerte f_j nach einem bestimmten Mechanismus, der durch charakteristische Parameter a_μ gekennzeichnet ist, erzeugt werden. Für eine andere Klasse hat man andere Parameter a_μ.

Die Bestimmung der Parameter a_μ erfolgt so, dass der durch

$$\varepsilon = \sum_{n=n_0}^{n_1} (f_n - \hat{f}_n)^2 \tag{3.6.4}$$

definierte Vorhersagefehler minimiert wird. Mit (3.6.3) erhält man als Bedingung

$$\frac{\delta\varepsilon}{\delta a_\nu} = 0 \;=\; \sum_n \left(f_n + \sum_\mu a_\mu f_{n-\mu} \right) 2 f_{n-\nu} \;, \tag{3.6.5}$$

$$\sum_\mu a_\mu \sum_n f_{n-\mu} f_{n-\nu} \;=\; -\sum_n f_n f_{n-\nu} \;, \quad \nu = 1, \ldots, m \;. \tag{3.6.6}$$

Mit (3.6.6) liegen m lineare Gleichungen zur Bestimmung der m Parameter a_μ vor. Die Art der Lösung hängt von den Annahmen über n_0, n_1 ab. Eine besonders effektive Lösung des Gleichungssystems ist mit der **Autokorrelationsmethode** möglich. Dabei setzt man $n_0 = -\infty,\ n_1 = \infty,\ f_n = 0$ für $n < 0$ und $n \geq M$. Als *Kurzzeit–Autokorrelationsfunktion* der Folge $[f_j]$ definiert man

$$r_{|\nu-\mu|} = \sum_{n=0}^{M-1-|\nu-\mu|} f_n \, f_{n+|\nu-\mu|} \;=\; \sum_n f_{n-\mu} f_{n-\nu} \;. \tag{3.6.7}$$

Damit lässt sich (3.6.6) auch in der Form

$$\sum_{\mu=1}^{m} a_\mu r_{|\nu-\mu|} = -r_\nu \;, \quad \nu = 1, \ldots, m \tag{3.6.8}$$

angeben. Die Matrix dieses Gleichungssystems ist eine TOEPLITZ-**Matrix**, wie man an der Umschreibung von (3.6.8) in

$$\begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{m-1} \\ r_1 & r_0 & r_1 & \cdots & r_{m-2} \\ \vdots & & & & \\ r_{m-1} & r_{m-2} & r_{m-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{pmatrix} \tag{3.6.9}$$

sieht. Daher lässt sich das Gleichungssystem rekursiv mit einer Komplexität von $\mathcal{O}(m^2)$ lösen. Der als LEVINSON-**Rekursion** bekannte Algorithmus berechnet lineare Vorhersagekoeffizienten $a_{i,\mu}$ sowie den Vorhersagefehler ε_i sukzessive für $i = 1, 2, \ldots, m$, $\mu = 1, 2, \ldots, i$. Die Koeffizienten $a_{i,\mu}$ ergeben eine lineare Vorhersage gemäß (3.6.3) mit i Koeffizienten. Die Rekursion verläuft in folgenden Schritten:

1. Man berechne $r_j,\ j = 0, 1, \ldots, m$ gemäß (3.6.7) und initialisiere

$$\varepsilon_0 = r_0\ , \quad k_1 = -\frac{r_1}{r_0}\ , \quad a_{1,0} = 1\ , \quad a_{1,1} = k_1\ , \quad \varepsilon_1 = \varepsilon_0(1 - k_1^2)\ .$$

2. Für $i = 1, \ldots, m-1$ führe man folgende Operationen aus:

$$\begin{aligned} k_{i+1} &= -\frac{1}{\varepsilon_i} \sum_{j=0}^{i} r_{|i+1-j|} a_{i,j}\ , && (3.6.10) \\ a_{i+1,0} &= 1\ , && (3.6.11) \\ a_{i+1,\mu} &= a_{i,\mu} + k_{i+1} a_{i,i+1-\mu} \quad , \quad \mu = 1, \ldots, i\ , && (3.6.12) \\ a_{i+1,i+1} &= k_{i+1}\ , && (3.6.13) \\ \varepsilon_{i+1} &= \varepsilon_i(1 - k_{i+1}^2)\ . && (3.6.14) \end{aligned}$$

3. Für $i = m - 1$ erhält man aus (3.6.12), (3.6.13) Koeffizienten $a_{m,\mu},\ \mu = 1, \ldots, m$, die gleich den linearen Vorhersagekoeffizienten a_μ in (3.6.3) sind, also Lösungen des Gleichungssystems (3.6.8), (3.6.9).

Die Koeffizienten k_i werden als *Reflektionskoeffizienten* bezeichnet. Der Fehler ε in (3.6.4) lässt sich in geschlossener Form angeben. Mit einem $(m+1)$ten Koeffizienten $a_0 = 1$ folgt aus (3.6.3), (3.6.4) mit (3.6.7) durch einfache Rechnung

$$\varepsilon = \sum_{\mu=0}^{m} \sum_{\nu=0}^{m} a_\mu a_\nu r_{|\mu-\nu|}\ ; \tag{3.6.15}$$

und mit (3.6.8)

$$\varepsilon = \sum_{\mu=0}^{m} a_\mu r_\mu\ . \tag{3.6.16}$$

Bei Anwendung der obigen Rekursionsgleichungen erhält man diesen Fehler direkt aus (3.6.14) zu

$$\varepsilon = \varepsilon_m\ . \tag{3.6.17}$$

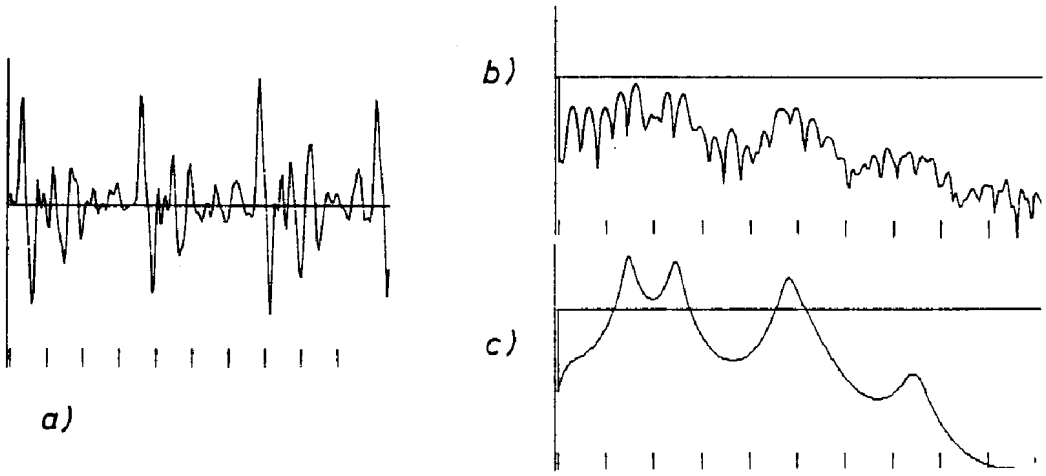


Bild 3.6.2: a) Eine Zeitfunktion; es handelt sich um einen Ausschnitt von 20 ms Dauer aus dem Vokal „a“ in dem Wort „Fass“. b) Das mit der DFT berechnete Spektrum, Abtastfrequenz 10kHz. c) Das aus den Koeffizienten der linearen Vorhersage mit $m = 13$ gewonnene Modellspektrum. Bei den Spektren ist jeweils der Betrag in logarithmischem Maßstab für den Bereich 0–5kHz dargestellt

Damit ist die Berechnung der linearen Vorhersagekoeffizienten nach der Autokorrelationsmethode abgeschlossen. Für ein Muster $f(t)$ wird die Rechnung meistens über kleine, sich etwas überlappende Zeitabschnitte ausgeführt.

Eine Möglichkeit besteht darin, die Vorhersagekoeffizienten direkt als Merkmale zu verwenden, wobei manchmal der Vorhersagefehler noch als weiteres Merkmal hinzugefügt wird, d. h. man setzt

$$\begin{aligned} c_\nu &= a_\nu\ , \quad \nu = 1, \ldots, m \\ c_{m+1} &= \varepsilon\ . \end{aligned} \tag{3.6.18}$$

Ebenso werden die Reflektionskoeffizienten (oder PARCOR–Koeffizienten) als Merkmale verwendet, d. h.

$$c_\nu = k_\nu\ , \quad \nu = 1, \ldots, m\ . \tag{3.6.19}$$

Man kann auch die Vorhersagekoeffizienten verwenden, um ein geglättetes **Modellspektrum** der Daten zu berechnen. Wenn die Abtastfrequenz für die Werte in der Folge $[f_j]$ mit f_s bezeichnet wird und die gewünschte Frequenzauflösung im Modellspektrum mit f_r, so wählt man

$$M' > \frac{f_s}{f_r} \tag{3.6.20}$$

und definiert einen Vektor $\boldsymbol{a}$ mit M' Elementen gemäß

$$\boldsymbol{a}_t = (1, a_1, a_2, \ldots, a_m, 0, 0, \ldots, 0)\ . \tag{3.6.21}$$

Einige oder alle Koeffizienten der DFT von $\boldsymbol{a}$ werden als Merkmale verwendet. Bei Anwendung der FFT nach Satz 3.3 muss zudem $M' = 2^q$ sein. Die Zahl m der Vorhersagekoeffizienten ist

problemabhängig, bei Sprache sind z. B. Werte $m = 10$ bis 15 üblich, oder in Abhängigkeit von der Abtastfrequenz f_s [kHz] des Sprachsignals $m = f_s + 4$ bis $m = f_s + 5$. Ein allgemeines Kriterium zur Wahl von m gibt (4.2.52), S. 337. Bild 3.6.2 zeigt ein Beispiel für ein FFT Spektrum und ein Modellspektrum. Das Modellspektrum ist eine Approximation des DFT–Spektrums, die immer besser wird je größer m wird. Die relativen Maxima im Modellspektrum entsprechen den *Formanten*, d. h. den Resonanzen im Vokaltrakt, die zahlreichen relativen Maxima im DFT–Spektrum werden durch die periodische Anregung (s. Bild 3.6.1) bei stimmhaften Lauten verursacht. Ein zu kleiner Wert von m führt zum Verschmelzen verschiedener Formanten, ein zu großer zum Durchschlagen der (aus Sicht der Spracherkennung uninteressanten) periodischen Anregung. Die genannten Richtwerte für m stellen einen sinnvollen Kompromiss dar.

Es ist zweckmäßig, vor den entsprechenden Rechnungen die Abtastwerte $[f_j]$ des Sprachsignals durch eine *Präemphase*

$$f'_j = \alpha f_j - (1-\alpha) f_{j-1} \,, \quad 0 \le \alpha \le 1 \tag{3.6.22}$$

vorzuverarbeiten, um den Einfluss der Glottisschwingung und der Lippenabstrahlung zu reduzieren. Dieses ist eine Hochpassfilterung im Sinne von Abschnitt 2.3.4.

3.6.3 Cepstrum Koeffizienten

Die Cepstrum Koeffizienten, speziell die mel–Cepstrum Koeffizienten, sind nach zahlreichen experimentellen Ergebnissen ein für die Spracherkennung sehr geeigneter Satz von Merkmalen. Sie sind daher seit mehreren Jahren *der* Ansatz für die Merkmalsgewinnung in der Spracherkennung. Eine Standardversion dieser Merkmale ist zusammenfassend wie folgt definiert:

Definition 3.15 *Die* **mel–Cepstrum Koeffizienten** *sowie ihre zeitlichen Ableitungen werden in den folgenden Schritten berechnet:*

1. *Berechnung der Koeffizienten* $c^{(\mathrm{ls})}_{\tau,k}$ *des* Leistungsspektrums *(oberer Index* ls*) für ein Datenfenster* $w_{\tau,j}$ *mit* N *Abtastwerten am diskreten Zeitpunkt* τ *gemäß (3.6.24).*
2. *Transformation in die Koeffizienten* $c^{(\mathrm{mf})}_{\tau,j}$ *der mel–Frequenzskala (oberer Index* mf*) gemäß (3.6.25), wobei* N_{d} *Filter* $d(j,k)$ *mit Mittenfrequenz* j *verwendet werden.*
3. *Berechnung von* N_{mc} *mel–Cepstrum Koeffizienten (oberer Index* mc*)* $c^{(\mathrm{mc})}_{\tau,k}$ *durch Logarithmierung und diskrete cosinus–Transformation (3.6.26).*
4. *Berechnung der ersten und zweiten zeitlichen (diskreten) Ableitungen* $\Delta c^{(\mathrm{mc})}_{\tau,k}$ *und* $\Delta\Delta c^{(\mathrm{mc})}_{\tau,k}$ *gemäß (3.6.27) und (3.6.28).*
5. *Bildung eines Merkmalsvektors* $\mathbf{c}_\tau$ *aus* $c^{(\mathrm{mc})}_{\tau,k}$, $\Delta c^{(\mathrm{mc})}_{\tau,k}$ *und* $\Delta\Delta c^{(\mathrm{mc})}_{\tau,k}$.
6. *Gegebenenfalls Kompression des (u. U. zu langen) Merkmalsvektors durch eine Hauptachsentransformation (s. Abschnitt 3.8.2).*

Freiheitsgrade bestehen z. B. in der Wahl der Art, Breite und zeitlichen Positionierung der Fensterfunktion w_j, der Wahl von Zahl, Mittenfrequenz und Breite der Dreiecksfenster sowie der Zahl der in den einzelnen Schritten verwendeten Koeffizienten. Statt Dreiecksfenstern können auch Rechteck– oder Trapezfenster verwendet werden. Weiterhin kann die Berechnung des Leistungsspektrums statt aus der diskreten FOURIER-Transformation aus dem Modellspektrum der linearen Vorhersage in Abschnitt 3.6.2 erfolgen. Die zeitlichen Ableitungen können wie in (3.6.27) durch Differenzen oder z. B. durch eine Regressionsgerade geschätzt

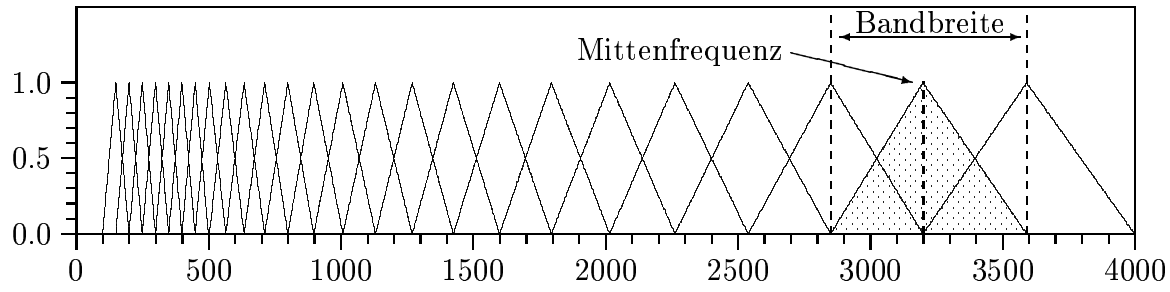


Bild 3.6.3: Eine Bank von Dreiecksfiltern; sieben Filter linear gestuft mit Mittenfrequenzen 150, 200, 250, ..., 400 Hz, je sechs Filter logarithmisch gestuft in den drei Oktaven 0,5 – 1 kHz, 1 – 2 kHz und 2 – 4 kHz. Jedes Band reicht von der Mittenfrequenz des linken zu der des rechten Nachbarfilters (mit Genehmigung des Autors aus [Schukat-Talamazzini, 1995])

werden. Schließlich kann die zeitliche Information auch durch Einbeziehung von Koeffizienten $c^{(\mathrm{mc})}_{\tau-i,k}$, $c^{(\mathrm{mc})}_{\tau-i+1,k}$, ..., $c^{(\mathrm{mc})}_{\tau,k}$ in den Merkmalsvektor berücksichtigt werden.

Die Berechnung der mel–Cepstrum Koeffizienten orientiert sich zunächst an dem Modell der Sprachproduktion in Bild 3.6.1. Danach wird das Anregungsignal mit der Impulsantwort des Vokaltrakts gefaltet. Für die Spracherkennung ist die zeitlich sich relativ rasch ändernde Anregung uninteressant, wichtig ist die im Vergleich dazu langsam veränderliche Änderung des Vokaltrakts, da diese den geformten Laut bestimmt. Als Merkmale sind daher vor allem die Cepstrum Koeffizienten niederer Ordnung relevant. Die Faltung wird, wie in (3.2.25) – (3.2.30), S. 172, gezeigt, durch Bildung des Cepstrums in eine additive Verknüpfung transformiert. Die in (3.2.30) verwendete Betragsbildung ist bei Sprache dadurch gerechtfertigt, dass die Phase für den auditiven Eindruck nicht relevant ist. Zudem erhält sie die Transformation von Faltung in Addition und erlaubt die Verwendung des reellen Logarithmus. Daher wird in (3.6.24) das Betragsquadrat der FOURIER-Koeffizienten verwendet.

Die Koeffizienten des Leistungsspektrum werden in (3.6.25) mit Dreiecksfiltern zusammengefasst. Diese orientieren sich zum einen an der von Versuchspersonen subjektiv empfundenen Tonhöhe, die als *Tonheit* bezeichnet und in der Einheit mel (melodische Tonheit) gemessen wird. Der Zusammenhang zwischen physikalischer Tonhöhe f_{Hz} [Hz] und Tonheit f_{mel} [mel] ist *nichtlinear*. Eine Approximation ist

$$f_{\mathrm{mel}} = 2595 \cdot \log\left[1 + \frac{f_{\mathrm{Hz}}}{700}\right] \quad . \tag{3.6.23}$$

Die Dreiecksfilter orientieren sich zum anderen an der Eigenschaft des menschlichen Ohres, die Lautstärke über *Frequenzgruppen* zu bilden, indem die Spektralanteile eines Frequenzbereichs bewichtet addiert werden. Der Frequenzbereich von 20 Hz bis 16 kHz wird von 24 *nichtüberlappenden* Frequenzgruppen überdeckt. Allerdings kann das Ohr an *jeder* Mittenfrequenz solche Gruppen bilden, sodass der Mensch (natürlich) mehr als 24 Tonhöhen unterscheiden kann. Für die Spracherkennung finden sich daher in der Literatur untschiedliche Zahl, Form und Frequenzaufteilungen für diese Filter. Ein Beispiel für die Filter $d_{l,k}$ zeigt Bild 3.6.3 in Form von $N_{\mathrm{d}} = 25$ Dreiecksfiltern.

Von den so gewonnenen Koeffizienten wird in (3.6.26) der Logarithmus verwendet. Dieses ist zum einen wiederum der Charakteristik des Ohres nachempfunden, zum anderen für die Berechnung des Cepstrums erforderlich. Zur Reduktion der durch die obigen Schritte gewonnenen

Zahl von Koeffizienten kann eine Hauptachsentransformation verwendet werden. Wie schon in Abschnitt 3.2.4 dargelegt, hat die diskrete cosinus Transformation ähnliche Eigenschaften wie die Hauptachsentransformation, ist aber schneller zu berechnen, da sie problemunabhängig ist. Daher wird in (3.6.26) die diskrete cosinus Transformation verwendet. Mit ihr werden N_{mc} mel–Cepstrum Koeffizienten berechnet, wobei in der Regel $20 - 30$ verwendet werden. Damit sind auch die Berechnungsschritte für das Cepstrum abgeschlossen, die nach der Logarithmierung eine weitere inverse DFT vorsehen; diese kann für reelle symmetrische Koeffizienten auch durch eine cosinus Transformation berechnet werden.

Die Information über einen Laut liegt nicht nur in den zu einem Zeitfenster vorliegenden Daten, sondern auch in deren zeitlicher Änderung. Daher ist es sinnvoll und verbessert die Erkennungsraten bei der Worterkennung, wenn die ersten und zweiten zeitlichen Ableitungen der Koeffizienten (3.6.27) und (3.6.28) zum Merkmalsvektor (3.6.29) hinzugefügt werden.

$$c_{\tau,k}^{(\mathrm{ls})} = \left| \frac{1}{N} \sum_{j=0}^{N-1} w_{\tau,j} f_j \exp\left[-\mathrm{i}\, 2\pi \frac{jk}{N} \right] \right|^2 , \quad k = 0, 1, \ldots, (N/2) - 1 , \tag{3.6.24}$$

$$c_{\tau,j}^{(\mathrm{mf})} = \sum_{k=0}^{(N/2)-1} d_{j,k}\, c_{\tau,k}^{(\mathrm{ls})} , \quad j = 1, \ldots, N_{\mathrm{d}} , \tag{3.6.25}$$

$$c_{\tau,k}^{(\mathrm{mc})} = \sum_{j=1}^{N_{\mathrm{d}}} \log\left[c_{\tau,j}^{(\mathrm{mf})} \right] \cdot \cos\left[\frac{k \cdot (2j-1)\pi}{2N_{\mathrm{d}}} \right] , \quad k = 1, \cdots, N_{\mathrm{mc}} \leq N_{\mathrm{d}} \tag{3.6.26}$$

$$\Delta c_{\tau,k}^{(\mathrm{mc})} = c_{\tau+1,k}^{(\mathrm{mc})} - c_{\tau-1,k}^{(\mathrm{mc})} , \tag{3.6.27}$$

$$\Delta\Delta c_{\tau,k}^{(\mathrm{mc})} = \Delta c_{\tau+1,k}^{(\mathrm{mc})} - \Delta c_{\tau-1,k}^{(\mathrm{mc})} , \tag{3.6.28}$$

$$\boldsymbol{c}_\tau = \left(c_{\tau,k}^{(\mathrm{mc})},\, \Delta c_{\tau,k}^{(\mathrm{mc})},\, \Delta\Delta c_{\tau,k}^{(\mathrm{mc})} , k = 1, \ldots, N_{\mathrm{mc}} \right)^{\mathsf{T}} . \tag{3.6.29}$$

Zwei Varianten der mel–Cepstrum Koeffizienten, die insbesondere für Spracherkennung unter Einfluss von Störgeräuschen Vorteile haben, sind die root-Cepstrum und die μ–Law Koeffizienten. Die **root–Cepstrum Koeffizienten** erhält man, indem man (3.6.26) ersetzt durch

$$c_{\tau,k}^{(\mathrm{rc})} = \sum_{j=1}^{N_{\mathrm{d}}} \left| c_{\tau,j}^{(\mathrm{mf})} \right|^r \cdot \cos\left[\frac{k \cdot (2j-1)\pi}{2N_{\mathrm{d}}} \right] , \quad r \approx 0,2 - 0,25 . \tag{3.6.30}$$

Die **μ–Law Koeffizienten** (oder mu–Law Koeffizienten) erhält man, indem man (3.6.26) ersetzt durch

$$c_{\tau,k}^{(\mathrm{muL})} = \sum_{j=0}^{N_{\mathrm{d}}} c_{\tau,\max}^{(\mathrm{mf})} \operatorname{sign}[c_{\tau,j}^{(\mathrm{mf})}] \frac{\log\left[1 + \mu |c_{\tau,j}^{(\mathrm{mf})}| / c_{\tau,\max}^{(\mathrm{mf})} \right]}{\log[1+\mu]} , \quad \mu \approx 10^5 - 10^7 . \tag{3.6.31}$$

Eine weitere Maßnahme zur Reduktion des Einflusses von Störgeräuschen ist die getrennte Berechnung der mel–Cepstrum Koeffizienten in unterschiedlichen Frequenzbändern.

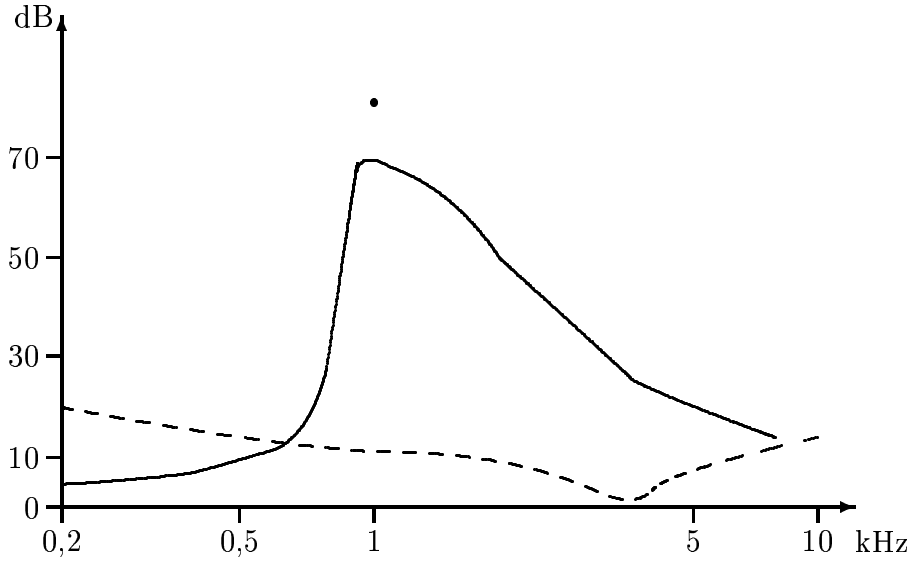


Bild 3.6.4: Verlauf der Maskierungskurve (durchgehende Linie) für einen 1 kHz Ton von 80 dB, angedeutet durch •; der Verlauf der Hörschwelle (gestrichelte Linie)

3.6.4 Lautheit

Die „Lautheit“ bzw. Energie eines Sprachsignals ist erfahrungsgemäß für die Spracherkennung wichtig. Als Maß dafür wird jedoch nicht die Energie am Ausgang der Dreiecksfilter (3.6.26) oder der nullte Koeffizient der DCT in (3.6.26) genommen, sondern ein daraus abgeleiteter Wert. Objektiv messbar ist der **Schallpegel**

$$L = 20 \log \left[\frac{p_s}{p_0}\right] = 10 \log \left[\frac{I_s}{I_0}\right] \tag{3.6.32}$$

mit der Einheit [dB]. Dabei ist p_s der messbare Schalldruck (in Pascal, Pa), I_s die Intensität [N m^{-2}] und $p_0 = 2 \cdot 10^{-5}$ Pa, $I_0 = 10^{-12}$ Nm^{-2} sind per Konvention festgelegte Bezugsgrößen. Der erforderliche Schallpegel für *subjektiv* mit gleicher Lautstärke wahrgenommene Töne wird mit Versuchspersonen ermittelt. Die maximale Empfindlichkeit des Ohres liegt danach bei etwa 4 kHz. Sie nimmt zu niedrigeren und höheren Frequenzen stark ab.

In Anlehnung daran wird als Maß für die Lautheit eine Größe wie

$$L_\tau^{\mathrm{mf}} \quad = \quad \sum_j 10 \log c_{\tau,j}^{(\mathrm{mf})} \tag{3.6.33}$$

verwendet. Diese wird oft als weitere Komponente zum Merkmalsvektor in (3.6.29) hinzugefügt.

3.6.5 Normierung

Zur Störungsreduktion wird oft die *spektrale Subtraktion* verwendet, die in (2.5.54), (2.5.55), S. 135, vorgestellt wurde. Eine weitere Maßnahme ist die Reduktion von Einflüssen des Mikrofons, des Raumes und der Sprechereigenschaften durch Normierung auf ein Langzeitspektrum. Eine Standardnormierung der Merkmale zu diesem Zweck ist der **cepstrale Mittelwertabzug**

$$c_{\tau,k}^{(\mathrm{mc,n})} = c_{\tau,k}^{(\mathrm{mc})} - m_k^{(\mathrm{mc})} \ , \tag{3.6.34}$$

wobei $m_k^{(\mathrm{mc})}$ der zeitliche Mittelwert des k–ten Cepstrum Koeffizienten und $c_{\tau,k}^{(\mathrm{mc,n})}$ der resultierende normierte Wert ist.

Ein anderer Vorschlag zur Normierung besteht aus den Schritten der Unterdrückung aller durch den Maskierungseffekt verdeckten Spektralkomponenten, der Lautstärkenormierung jeder Spektralkomponente mit der Lautstärke an der Hörschwelle sowie der Abbildung in die mel–Skala mit (3.6.23). Der **Maskierungseffekt** besagt, dass alle Geräusche *unhörbar* werden, wenn andere lautere zeitlich und spektral benachbart sind. Der ungefähre Verlauf der Maskierung geht aus Bild 3.6.4a hervor. Zu jeder Spektralkomponente f_ν des Sprachsignals wird diese Maskierung, vertikal verschoben je nach aktueller Lautstärke, angenommen und ihr Verlauf über der Signalfrequenz f_s mit $\psi_\nu(f_\mathrm{s})$ bezeichnet. Für das aktuelle Sprachsignal zur (diskreten) Zeit τ ist die resultierende Maskierung dann definiert mit

$$\psi_\tau(f_\mathrm{s}) = \max_\nu\{\psi_\nu(f_\mathrm{s})\} \; . \tag{3.6.35}$$

Alle Anteile des Sprachsignals *unterhalb* der Maskierung $\psi_\tau(f_\mathrm{s})$ werden eliminiert. Der Verlauf der **Hörschwelle** ist in Bild 3.6.4b angegeben. Ist $L_\mathrm{hs}(f_\mathrm{s})$ die Lautstärke an der Hörschwelle und $L_\mathrm{s}(f_\mathrm{s})$ die Lautstärke des Sprachsignals, so ist die normierte Lautstärke

$$L_\mathrm{sn}(f_\mathrm{s}) = L_\mathrm{s}(f_\mathrm{s}) - L_\mathrm{hs}(f_\mathrm{s}) \; . \tag{3.6.36}$$

3.7 Merkmale für die Objekterkennung (VA.1.1.2, 14.05.2004)

3.7.1 Vorbemerkung

Unterschiedliche Klassifikationsprobleme, darunter auch die Klassifikation dreidimensionaler Objekte aus einer zweidimensionalen Ansicht, wurden bereits in Abschnitt 1.5 vorgestellt und illustriert. Grundsätzlich kommen dafür die in den vorigen Abschnitten dieses Kapitels eingeführten Verfahren der Merkmalsgewinnung in Frage. Es wurden jedoch auch Merkmale speziell für dieses Problem vorgeschlagen, sodass es angemessen ist, hier einige davon vorzustellen.

3.7.2 Lokale Merkmale

Die Gewinnung von Merkmalen gemäß (3.1.1), S. 163, oder speziell (3.2.2), S. 166, bedeutet, dass ein Muster als Ganzes transformiert wird. Änderungen in einem kleinen Bereich des Musters wirken sich dann i. Allg. auf *alle* Koeffizienten des Merkmalsvektors aus. Allerdings wurde bereits mehrfach auf die Blockbildung oder Fensterung bei der Merkmalsgewinnung hingewiesen. Hier werden speziell *lokale Merkmale* betrachtet, die nur in einer begrenzten Umgebung eines Bildpunktes berechnet werden. Dieses kann durch gefensterte Transformationen erfolgen (z. B. die gefensterte FOURIER-Transformation, s. Abschnitt 3.2.3), durch Verwendung von Funktionen, die nur in einem endlichen Bereich definiert sind (z. B. die Wavelets, s. Abschnitt 3.3) bzw. außerhalb eines endlichen Bereichs sehr rasch abklingen (z. B. die GABOR-Funktionen s. Abschnitt 3.4.1) sowie i. Allg. durch lineare Filter- oder Maskenoperationen (s. Abschnitt 2.3.2 und Abschnitt 2.3.5). Ihr Vorteil ist, dass sich lokale Verdeckungen, Verzerrungen oder Störungen im Muster nur lokal auf die Merkmale auswirken.

Es wird hier davon ausgegangen, dass lokale Merkmale an gleichmäßig über das Bild verteilten Positionen berechnet werden. Diese kann man z. B. gewinnen, indem man einen n-dimensionalen *lokalen Merkmalsvektor* $\boldsymbol{c}_{j',k'}$ an Positionen

$$\begin{aligned} j' &= j_0 + \mu\Delta x_c\,, \quad k' = k_0 + \nu\Delta y_c)\,, \\ &\mu = 0,1,\ldots,\lfloor (M_x - 2j_0)/\Delta x_c \rfloor,\ \nu = 0,1,\ldots,\lfloor (M_y - 2k_0)/\Delta y_c \rfloor \end{aligned} \tag{3.7.1}$$

berechnet. Dabei ist $\lfloor a \rfloor$ die größte ganze Zahl, die nicht größer als a ist. Der Abstand und die Zahl der lokalen Merkmale ist also durch die Zahl $M_x \times M_y$ der Abtastwerte des Bildes, den Randabstand (j_0, k_0) und die Schrittweite $\Delta x_c, \Delta y_c$ bestimmt; diese Schrittweite ist nicht zu verwechseln mit den in (2.1.1), S. 62, eingeführten Abständen $(\Delta x, \Delta y)$ der Abtastwerte. Der Spezialfall, dass ein lokaler Merkmalsvektor je Bildpunkt berechnet wird, ist darin enthalten mit $j_0 = k_0 = 0$, $\Delta x_c = \Delta y_c = \Delta x = \Delta y = 1$. Die diskreten Positionen (j', k') werden zur Vereinfachung der Notation auch mit nur einem Index m numeriert, ihre Position auch mit $\boldsymbol{x}_m$ angegeben. Man erhält N_c lokale Merkmalsvektoren $\boldsymbol{c}_{j',k'} = \boldsymbol{c}(\boldsymbol{x}_m) = \boldsymbol{c}_m$; sie ergeben die Menge

$$\begin{aligned} \widetilde{C} &= \{\boldsymbol{c}_m\} = \{(c_{m,1},\ldots,c_{m,\nu},\ldots,c_{m,n})^\mathsf{T}\}\,, \quad m = 1,2,\ldots,N_c\,, \\ N_c &= (\lfloor (M_x - 2j_0)/\Delta x_c \rfloor + 1)\cdot(\lfloor (M_y - 2k_0)/\Delta y_c \rfloor + 1)\ . \end{aligned} \tag{3.7.2}$$

In (2.4.14), S. 111, wurden Tupel $(j, k, f_{jk})^\mathsf{T}$ betrachtet; hier werden analog Tupel $(j', k', \boldsymbol{c}_{j',k'})^\mathsf{T}$ von lokalen Merkmalen $\boldsymbol{c}_{j',k'} = \boldsymbol{c}_m$ betrachtet.

Lineare Filter

(Nicht rekursive) lineare Filteroperationen werden durch eine diskrete *Faltung* mit einer Impulsantwort $g_{\mu\nu}$ realisiert (s. (2.3.15), S. 89), die auch als Multiplikation eines Ausschnittes des Bildes mit einer geeigneten *Maske* $\bar{g}_{\mu\nu}$ realisiert werden kann (s. (3.5.24), S. 205). Eine solche Maskenoperation ergibt für ein Bild $[f_{jk}]$

$$h_{jk} = \sum_{\mu=-m_x}^{m_x} \sum_{\nu=-m_y}^{m_y} f_{j+\mu,k+\nu}\, \bar{g}_{\mu\nu} \ . \tag{3.7.3}$$

In der Literatur wird empfohlen, diese Werte zur Elimination der Intensitätsabhängigkeit geeignet zu normieren. Als zusätzlich besonders robust gegenüber additivem GAUSS-Rauschen hat sich die Energienormierung

$$h_{jk} = \frac{\sum_{\mu=-m_x}^{m_x} \sum_{\nu=-m_y}^{m_y} f_{j+\mu,k+\nu}\, \bar{g}_{\mu\nu}}{\sqrt{\sum_{\mu=-m_x}^{m_x} \sum_{\nu=-m_y}^{m_y} f_{j+\mu,k+\nu}^2}\ \sqrt{\sum_{\mu=-m_x}^{m_x} \sum_{\nu=-m_y}^{m_y} \bar{g}_{\mu\nu}^2}} \tag{3.7.4}$$

erwiesen.

Wenn man n lineare Operationen $\boldsymbol{g}^{(\nu)}$, $\nu = 1, \ldots, n$ vorgibt und ihr Ergebnis an N_c ausgewählten Positionen (j', k') im Bild berechnet, erhält man je Position, die wir wie oben zur Vereinfachung mit m indizieren, einen n-dimensionalen Merkmalsvektor $\boldsymbol{c}_m$ mit Komponenten $c_{m,\nu}, \nu = 1, \ldots, n$.

Beispiele für lineare Operationen sind Ableitungen der Grauwertfunktion (s. (3.7.5) und (3.7.6)), GAUSS-Filter mit zwei oder drei Werten für die Varianz σ und deren Richtungsableitungen nach x und y (s. Abschnitt 2.3.5 sowie Abschnitt 2.3.6 zur rekursiven Realisierung), Wavelet–Transformation (s. Abschnitt 3.3.4) oder GABOR-Filter (s. Abschnitt 3.4.1). Es werden auch Merkmale direkt aus dem Bild $\boldsymbol{f}$ berechnet. Masken für Richtungsableitungen der Grauwertfunktion und für den LAPLACE-Operator sind z. B.

$$\bar{g}^{(x)} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} , \quad \bar{g}^{(y)} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} , \tag{3.7.5}$$

$$\bar{g}^{(\mathrm{L})} = \begin{pmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{pmatrix} . \tag{3.7.6}$$

Nutzung von Wavelets

Ein Beispiel für lokale Merkmalsvektoren erhält man durch eine Wavelet–Transformation, für die man zweckmäßigerweise $\Delta x_c = \Delta y_c = 2^s$ setzt und z. B. $s = 2$ wählt. Die zweidimensionale Wavelet–Transformation wird auf eine Nachbarschaft von $2^s \times 2^s$ Bildpunkten um die Position (j', k') angewendet. Wenn man sie s–mal ausführt, erhält man an jeder Position $(j', k') = m$ einen Mittelwertkoeffizienten (oder Tiefpassanteil) $f_{s,m}$ und drei Koeffizienten $d_{0,s,m}, d_{1,s,m}, d_{2,s,m}$, die aus der Kombination horizontaler und vertikaler Hoch- und Tiefpassanteile entstehen, wie auch Bild 3.3.6, S. 195, zeigt. Daraus wird ein zweidimensionaler Merkmalsvektor je Position berechnet

$$\boldsymbol{c}_m = \begin{pmatrix} c_{m,1} \\ c_{m,2} \end{pmatrix} = \begin{pmatrix} \ln\left[2^{-s}|f_{s,m}|\right] \\ \ln[2^{-s}(|d_{0,s,m}| + |d_{1,s,m}| + |d_{2,s,m}|)] \end{pmatrix} . \tag{3.7.7}$$

Wie in der Literatur ausführlich diskutiert wird, wird durch die Logarithmierung der Wavelet–Koeffizienten die Abhängigkeit von der Beleuchtung reduziert; durch die Addition der Hochpassanteile wird die Abhängigkeit von Rotationen reduziert; durch die Multiplikation mit dem Faktor 2^{-s} wird der Wertebereich unabhängig von der Zahl s der Transformationsschritte. Um zu kleine Werte zu vermeiden, kann noch die Logarithmierung auf Werte größer $\ln[0,5] \approx -0,69$ beschränkt werden. Von den zahlreichen Wavelets hat sich in Klassifikationsexperimenten das JOHNSTON 8-Wavelet (s. Tabelle 3.1, S. 195) als günstig erwiesen, das dem HAAR-Wavelet ähnlich ist, aber einen größeren Einflussbereich hat.

3.7.3 Kombinierte Merkmale

Merkmale auf der Basis von linearen Filtern und Reihenentwicklungen gehören zum Standardinventar. Als Alternative geben wir noch einen Merkmalssatz an, der unterschiedliche Elemente, nämlich Farb–, Form–, Kontur- und GABOR-Filterwerte, kombiniert. Die Kamerabilder haben die Auflösung 480×480 Bildpunkte.

Das Kamerabild wird durch Tiefpassfilterung und Unterabtastung auf die Größe 120×120 reduziert. Es werden 22 *Farbkriterien*, im H, S, I–Farbraum berechnet. Dieser wird aus R, G, B Werten, die auf das Intervall $[0, 1]$ normiert sind, berechnet

$$\begin{aligned} H &= \arctan\left[\frac{\sqrt{3}(G-B)}{2R-G-B}\right] , \\ S &= 1 - \frac{\min\{R,G,B\}}{I} , \\ I &= \frac{R+G+B}{3} . \end{aligned} \tag{3.7.8}$$

Im Kreis der Farbtöne H werden beginnend bei 0^o 11 äquidistante Zentren definiert. Je Zentrum wird eine symmetrische Dreiecksfunktion eingeführt, die im Zentrum den Wert $H_z = 1$ liefert und dann innerhalb 45^o linear auf $H_z = 0$ abfällt. Mit der Farbsättigung S werden dann zwei Werte h^{h} (hohe Sättigung) und h^{g} (geringe Sättigung) in jedem der 11 Zentren berechnet gemäß

$$h^{\mathrm{h}} = H_z \Theta(S,\, 0.4,\, 10)\Theta(I,\, 0.2,\, 20) , \tag{3.7.9}$$

$$h^{\mathrm{g}} = H_z \left(1 - \Theta(S,\, 0.4,\, 10)\right) \Theta(S,\, 0.15,\, 20)\Theta(I,\, 0.2,\, 20) , \tag{3.7.10}$$

$$\Theta(x, x_0, \beta) = \frac{1}{1 + \exp[\beta(x_0 - x)]} .$$

Dazu kommt noch ein Weißwert

$$h^{\mathrm{w}} = \Theta(I,\, 0.6,\, 20)\left(1 - \Theta(S,\, 0.15,\, 20)\right) . \tag{3.7.11}$$

Die 23 Werte werden für jeden Bildpunkt berechnet und jeder Wert über 0.1 führt zur Addition einer Eins in einem Histogramm mit 23 Einträgen.

Das Kamerabild wird durch Tiefpassfilterung und Unterabtastung auf die Größe 120×120 reduziert. Zu den obigen Merkmalen kommen 11 hinzu, die Ecken mit *Öffnungswinkeln* von 30^o, 60^o, ..., 330^o charakterisieren. Die Kanten der Ecken müssen entlang der Kante hinreichend homogen und senkrecht zur Kante hinreichend kontrastreich sein. Es werden 12 Kanten mit Orientierungen 0^o, 30^o, 60^o, ..., 330^o untersucht. Entlang einer Kante werden je drei Bildpunkte $f_{\mathrm{l},i}$, $i = 1, 2, 3$ auf der einen und drei Bildpunkte $f_{\mathrm{r},i}$, $i = 1, 2, 3$ auf der anderen Seite

der Kante genommen. Sie erstrecken sich im Ausgangsbild über etwa 20 Bildpunkte. Die Bewertung einer Kante wird berechnet aus

$$y = \prod_{i=1}^{3} \Theta(|f_{\mathrm{l},i} - f_{\mathrm{r},i}|,\, 0.08,\, 30) - 0.8 \sum_{i=1}^{3} \left(|f_{\mathrm{l},i} - m_{\mathrm{l}}| + |f_{\mathrm{r},i} - m_{\mathrm{r}}|\right) \ . \tag{3.7.12}$$

Dabei ist m_{l} bzw. m_{r} die mittlere Bildhelligkeit auf der einen bzw. der anderen Seite der Kante. Das Ergebnis wird quantisiert, indem Werte $y > 0$ auf Eins, sonst auf Null gesetzt werden. Je Bildpunkt hat man damit eine Aussage, ob Kanten vorliegen und wenn ja, mit welchen Orientierungen. Eine Ecke mit einem der 11 Öffnungswinkel liegt in einem Bildpunkt vor, wenn es dort zwei Kanten mit entsprechender *relativer* Orientierung gibt; die *absolute* Orientierung spielt keine Rolle. Wenn eine Ecke vorliegt, wird eine Eins in einem Histogramm mit 11 Einträgen addiert.

Weiterhin werden 12 Typen von *Intensitätsflecken* berechnet. Es werden drei Bilder mit den Auflösungen 120×120, 60×60 und 30×30 verwendet. Alle Operationen werden auf jedem der drei Bilder ausgeführt. Ein Intensitätsfleck ist eine Menge von Intensitätsgradienten mit konsistentem Vorzeichen in elliptischer Konfiguration um ein Zentrum. Die Vorgehensweise ist analog zu der bei der Berechnung der Intensitätsgradienten zur Gewinnung der Ecken, jedoch werden die Intensitätsdifferenzen nun nicht von Punktepaaren entlang einer Kante, sondern um eine elliptische Region berechnet. Es wird auch nicht wie in (3.7.12) das Produkt von drei Intensitïsdifferenzen gebildet, sondern die Summe von zwölf solchen Termen entlang einer elliptischen Kontur. Ein Homogenitätsterm wird analog zu (3.7.12) berechnet und subtrahiert. Wenn das Maß den Schwellwert 4 übersteigt, wird ein Histogrammzähler erhöht. Die 12 Typen von Intensitätsflecken sind helle und dunkle, runde und längliche Flecken in den drei Auflösungen.

Es werden 40 *Konturmerkmale* berechnet. Auf dem Kamerabild wird eine Tiefpassfilterung vorgenommen, aber keine Unterabtastung, d. h. die Bildgröße ist 480×480 Bildpunkte. Kanten werden aus fünf Paaren von gegenüberliegenden Punkten, die etwa eine Länge von 18 Bildpunkten überspannen, in Schritten von $7,5^o$ berechnet. Eine Kante wird angenommen, wenn alle fünf Gradienten einen Schwellwert überschreiten, der vom Gradienten zwischen den Bildpunkten an den Kantenenden abhängt; dieses wird in den R, G, B Farbkanälen geprüft. Damit erhält man 24 Konturkarten im Winkelbereich 0^o bis $172,5^o$ mit einer Auflösung von 240×240 Bildpunkten. Daraus werden 40 Konturmerkmale gebildet, die aus sechs Kanten mit bestimmter Lage und Orientierung bestehen. Wenn wenigstens fünf der Kanten vorhanden sind, wird ein Histogrammzähler erhöht. Die Konturmerkmale sind 7 lange, einfache Konturtypen mit unterschiedlicher Krümmung, 9 Eckenmerkmale mit Öffnungswinkeln zwischen 30^o und 150^o sowie 24 Kantenpaare mit unterschiedlichem Abstand.

Schließlich kommen 16 Texturmerkmale dazu, die auf 40 GABOR-Filtern mit 8 Orientierungen und 5 Skalierungen (mit Abstufung $\sqrt{2}$) beruhen. Es wird je Bildpunkt das Betragsquadrat der Filterausgabe berechnet und als Energie bezeichnet. Die ersten 5 Texturkanäle bestehen aus der Summe der Energiewerte über alle Orientierungen je Skalierung, die nächsten 5 Texturkanäle aus der Varianz je Skalierung. Die letzten 6 Texturkanäle messen die Wahrscheinlichkeit relativer Energiewerte für Orientierungsdifferenzen von 0^o, 45^o, 90^o und Abstände von mehr bzw. weniger als 30 Bildpunkten. Histogrammeinträge werden erhöht, wenn die Ausgabewerte einen Schwellwert übersteigen.

3.8 Analytische Methoden (VA.1.2.2, 10.01.2004)

3.8.1 Kriterien

Als analytische Methoden zur Gewinnung von Merkmalen werden hier solche bezeichnet, mit denen man nach Vorgabe eines Kriteriums zur Bewertung der Güte der Merkmale systematisch genau die n Merkmale $c_\nu,\ \nu = 1, \ldots, n$ extrahieren kann, die das Kriterium maximieren (oder minimieren). Die Realisierung dieser Idee ist jedoch nur unter einschränkenden Annahmen möglich.

In Abschnitt 3.1 wurde der allgemeine Ansatz gemacht, dass Merkmale sich mit (3.1.1) aus einer Transformation

$$\boldsymbol{c} = T_{\mathrm{r}}\{\boldsymbol{f}\}$$

ergeben. Nach Vorgabe eines Kriteriums ist die das Kriterium optimierende Transformation T zu bestimmen. Das ist aber (bisher?) nicht möglich, ohne die Klasse der zulässigen Transformationen einzuschränken. Mathematisch ist zur Zeit praktisch nur gemäß (3.2.2) die Klasse der *linearen Transformationen*

$$\boldsymbol{c} = \boldsymbol{\Phi}\,\boldsymbol{f}$$

untersucht worden, und auch hier werden ausschließlich diese behandelt. Die in Abschnitt 3.8.3 behandelte nichtlineare Transformation ist zwar *nichtlinear* in $\boldsymbol{f}$, jedoch *linear* in den zunächst noch unbekannten Elementen der Matrix $\boldsymbol{\Phi}$; letzteres ist entscheidend für die einfache Berechnung der Elemente der Transformationsmatrix. Die letzte Gleichung wurde bereits in Abschnitt 3.2.1 eingeführt, jedoch soll $\boldsymbol{\Phi}$ hier nicht notwendig eine Orthogonaltransformation kennzeichnen; allerdings werden nichtorthogonale Transformationen erst in Abschnitt 3.8.4 erörtert.

Eine analytische Methode zur Merkmalsgewinnung erfordert demnach die Berechnung einer Matrix $\boldsymbol{\Phi}$, sodass die Merkmale $\boldsymbol{c} = \boldsymbol{\Phi}\boldsymbol{f}$ ein **Gütekriterium** optimieren. Es werden hier folgende Arten von Kriterien erwähnt:

1. Kriterien, welche auf Postulat 3 (vgl. S. 20) basieren, d. h. es werden quantitative Ausdrücke zur Bewertung der Konzentration von Merkmalen einer Klasse und der Trennung von Merkmalen verschiedener Klassen angegeben.
2. Kriterien, welche auf der Fehlerwahrscheinlichkeit des Klassifikators oder einer Abschätzung derselben basieren.
3. Kriterien, die auf dem Optimierungskriterium für den verwendeten Klassifikator beruhen, falls dieses nicht die Fehlerwahrscheinlichkeit ist.

Beispiele für die ersten beiden Kriterien werden in den folgenden Abschnitten gegeben. Insbesondere für Kriterien der ersten Art lassen sich mit Hilfe orthonormaler Basisvektoren mathematisch geschlossene Lösungen zur Berechnung der Transformationsmatrix $\boldsymbol{\Phi}$ angeben. Kriterien der zweiten Art sind dem Zweck des Klassifikationssystems direkt angepasst und daher an sich eindeutig den ersteren vorzuziehen. Die tatsächliche Berechnung der Transformationsmatrix ist jedoch erheblich schwieriger und erfordert vielfach weitere einschränkende Annahmen. Ein Kriterium der dritten Art wird erst in Abschnitt 4.4.3 in Zusammenhang mit Bild 4.4.1, S. 379 erwähnt, da dort der betreffende Klassifikator vorgestellt wird.

3.8.2 Problemabhängige Reihenentwicklung

Abstandskriterien

Wie in Abschnitt 3.2.1 beschränken wir uns in diesem Abschnitt auf lineare orthogonale Transformationen, d. h. das Muster wird nach einem orthonormalen Basisvektorsystem entwickelt. Im Unterschied zu Abschnitt 3.2.1 soll hier jedoch dasjenige System $\boldsymbol{\varphi}_\nu,\ \nu = 1,\ldots,n$ bestimmt werden, das ein geeignet gewähltes Kriterium optimiert. Es wird sich zeigen, dass in diesem Falle die Vektoren $\boldsymbol{\varphi}_\nu$ von den Mustern selbst, genauer von einer Stichprobe ω von Mustern abhängen, also je nach Problem verschieden sind. Daher wird eine solche Entwicklung auch als *problem***ab***hängige Reihenentwicklung* bezeichnet. Die im Abschnitt 3.2 vorgestellten Entwicklungen sind dagegen von den Mustern unabhängig, sie werden daher auch als *problem***unab***hängige Entwicklungen* bezeichnet.

Zunächst sind also geeignete Kriterien anzugeben, um ein „optimales“ Vektorsystem zu berechnen. Wenn man Muster ${}^j\boldsymbol{f}$ entwickelt, so mag auf den ersten Blick der Erwartungswert des quadratischen Approximationsfehlers als geeignetes Kriterium erscheinen. Es wurde aber bereits darauf hingewiesen, dass es weniger auf gute Approximation als auf sichere Klassifikation ankommt. Daher wird dieses Kriterium hier nicht weiter betrachtet. Statt dessen werden, wie schon im vorigen Abschnitt erwähnt, Kriterien verwendet, die auf Postulat 3 von Abschnitt 1.3 beruhen. Da man den aus einem Muster extrahierten Merkmalsvektor als Punkt im Merkmalsraum auffassen kann, ergibt eine Stichprobe von Mustern eine Punktmenge in diesem Raum. Postulat 3 besagt, dass die Punkte (Muster) der gleichen Klasse dicht beisammen liegen sollen, die verschiedener Klassen weit auseinander. Es ist intuitiv einleuchtend, dass die Klassifikation dann besonders einfach ist. Wenn man ein Abstandsmaß definiert, so lassen sich verschiedene Punktmengen quantitativ vergleichen und die beste – im Sinne von Postulat 3 – ermitteln. Als Maß für den Abstand zweier Merkmalsvektoren wird das Quadrat des EUKLID-Abstands gewählt. Damit werden die folgenden Kriterien zur Beurteilung einer Menge von Merkmalen $\omega = \{{}^j\boldsymbol{c} \mid j = 1,\ldots,N\}$ angegeben:

1. Mittlerer quadratischer *Abstand aller Merkmale* von allen anderen, definiert durch

$$s_1 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left({}^i\boldsymbol{c} - {}^j\boldsymbol{c}\right)^{\mathsf{T}}\left({}^i\boldsymbol{c} - {}^j\boldsymbol{c}\right)\ . \tag{3.8.1}$$

2. Mittlerer quadratischer Abstand aller Merkmale $\{{}^j\boldsymbol{c}_\kappa \mid j = 1,\ldots,N_\kappa\}$ aus einer Klasse Ω_κ von den Merkmalen einer anderen Klasse Ω_λ **(Interklassenabstand)**, definiert durch

$$s_2 = \frac{2}{k(k-1)}\sum_{\kappa=2}^{k}\sum_{\lambda=1}^{\kappa-1}\frac{1}{N_\kappa N_\lambda}\sum_{i=1}^{N_\kappa}\sum_{j=1}^{N_\lambda}\left({}^i\boldsymbol{c}_\kappa - {}^j\boldsymbol{c}_\lambda\right)^{\mathsf{T}}\left({}^i\boldsymbol{c}_\kappa - {}^j\boldsymbol{c}_\lambda\right)\ . \tag{3.8.2}$$

   Mit der ersten Doppelsumme werden alle verschiedenen Paare von Klassen erfasst, nämlich $\binom{k}{2} = k(k-1)/2$, mit der zweiten alle Abstände zwischen je einem Merkmalsvektor aus je einer Klasse des betreffenden Klassenpaares. Dabei ist k, wie üblich, die Zahl der Klassen.

3. Mittlerer quadratischer Abstand von Merkmalen der gleichen Klasse **(Intraklassenabstand)**, definiert durch

$$s_3 = \frac{1}{k}\sum_{\kappa=1}^{k}\frac{1}{N_\kappa^2}\sum_{i=1}^{N_\kappa}\sum_{j=1}^{N_\kappa}\left({}^i\boldsymbol{c}_\kappa - {}^j\boldsymbol{c}_\kappa\right)^{\mathsf{T}}\left({}^i\boldsymbol{c}_\kappa - {}^j\boldsymbol{c}_\kappa\right) \ . \tag{3.8.3}$$

4. Mittlerer quadratischer Abstand der *Klassenzentren* bzw. der bedingten Mittelwerte $\boldsymbol{\mu}_\kappa$ der Merkmale $\boldsymbol{c}_\kappa$

$$s_4 = \frac{2}{k(k-1)}\sum_{\kappa=2}^{k}\sum_{\lambda=1}^{\kappa-1}\left(\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda\right)^{\mathsf{T}}\left(\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda\right) \ . \tag{3.8.4}$$

Für die Klassifikation ist es günstig, wenn s_1, s_2 und s_4 *groß* sind und wenn s_3 *klein* ist. Da $\boldsymbol{c}$ mit (3.2.2) von $\boldsymbol{\Phi}$ abhängt, ist

$$s_l = s_l(\boldsymbol{\Phi}) \ , \qquad l = 1, \ldots, 4 \ . \tag{3.8.5}$$

Gesucht wird die Transformationsmatrix $\boldsymbol{\Phi}^{(l)}$, die für eine vorgegebene Merkmalszahl, d. h. für ein bestimmtes n, s_l optimiert. Zum Beispiel muss gelten, dass s_1 bezüglich $\boldsymbol{\Phi}$ *maximiert* wird.

Berechnung der Merkmale

Die Berechnung der im Sinne obiger Kriterien optimalen Merkmale lässt sich auf das bekannte Problem der Maximierung (bzw. Minimierung) einer quadratischen Form zurückführen. Das wesentliche Ergebnis wird zunächst in einem Satz zusammengefasst. Für die Berechnung der Transformationsmatrizen gilt:

Satz 3.8 *Die Transformationsmatrix, die das Kriterium s_l, $l = 1, \ldots, 4$ optimiert, werde mit $\boldsymbol{\Phi}^{(l)}$ bezeichnet. Man erhält $\boldsymbol{\Phi}^{(l)}$ indem man die* **Eigenvektoren** *$\boldsymbol{\varphi}_\nu^{(l)}$ einer geeigneten symmetrischen Kernmatrix $\boldsymbol{Q}^{(l)}$ berechnet, d. h. die Gleichung*

$$\boldsymbol{Q}^{(l)}\boldsymbol{\varphi}_\nu^{(l)} = \lambda_\nu^{(l)}\boldsymbol{\varphi}_\nu^{(l)} \tag{3.8.6}$$

löst, wobei die $\lambda_\nu^{(l)}$ die **Eigenwerte** *von $\boldsymbol{Q}^{(l)}$ sind. Zur Maximierung von s_1 bzw. s_2 sind die n Eigenvektoren $\boldsymbol{\varphi}_\nu^{(1)}$ bzw. $\boldsymbol{\varphi}_\nu^{(2)}$ zu berechnen, die zu den n größten Eigenwerten $\lambda_\nu^{(1)}$ bzw. $\lambda_\nu^{(2)}$, $\nu = 1, \ldots, n$ der Kerne $\boldsymbol{Q}^{(1)}$ bzw. $\boldsymbol{Q}^{(2)}$ gehören. Zur Minimierung von s_3 sind entsprechend die zu den kleinsten Eigenwerten von $\boldsymbol{Q}^{(3)}$ gehörigen Eigenvektoren zu bestimmen. Die so ermittelten n Eigenvektoren $\boldsymbol{\varphi}_\nu^{(l)}$ werden den n Zeilen von $\boldsymbol{\Phi}^{(l)}$ zugeordnet.*

Die Transformationsmatrix ist also

$$\boldsymbol{\Phi}^{(l)} = \begin{pmatrix} \boldsymbol{\varphi}_1^{(l)\mathsf{T}} \\ \boldsymbol{\varphi}_2^{(l)\mathsf{T}} \\ \vdots \\ \boldsymbol{\varphi}_n^{(l)\mathsf{T}} \end{pmatrix} \ . \tag{3.8.7}$$

Mit (3.2.2) ergeben sich dann aus einem Muster ${}^j\boldsymbol{f}$ die n Merkmale ${}^jc_\nu,\ \nu = 1,\ldots,n$. Bevor unten am Beispiel von s_2 der Beweis zu obigem Satz geführt wird, werden der Vollständigkeit halber noch die Kernmatrizen angegeben. Die Kerne sind definiert durch

$$\boldsymbol{Q}^{(1)} = \boldsymbol{R} - \boldsymbol{m}\boldsymbol{m}^{\mathsf{T}} , \tag{3.8.8}$$

$$\boldsymbol{R} = \frac{1}{N}\sum_{j=1}^{N} {}^j\boldsymbol{f}\,{}^j\boldsymbol{f}^{\mathsf{T}} , \quad \boldsymbol{m} = \frac{1}{N}\sum_{j=1}^{N} {}^j\boldsymbol{f} , \quad {}^j\boldsymbol{f} \in \omega , \tag{3.8.9}$$

$$\boldsymbol{Q}^{(2)} = \frac{1}{k}\sum_{\kappa}^{k} \boldsymbol{R}_\kappa - \frac{1}{k(k-1)}\sum_{\kappa=2}^{k}\sum_{\lambda=1}^{\kappa-1}(\boldsymbol{m}_\kappa\boldsymbol{m}_\lambda^{\mathsf{T}} + \boldsymbol{m}_\lambda\boldsymbol{m}_\kappa^{\mathsf{T}}) , \tag{3.8.10}$$

$$\boldsymbol{R}_\kappa = \frac{1}{N_\kappa}\sum_{j=1}^{N_\kappa} {}^j\boldsymbol{f}_\kappa\,{}^j\boldsymbol{f}_\kappa^{\mathsf{T}} , \quad \boldsymbol{m}_\kappa = \frac{1}{N_\kappa}\sum_{j=1}^{N_\kappa} {}^j\boldsymbol{f}_\kappa , \qquad {}^j\boldsymbol{f}_\kappa \in \omega_\kappa , \tag{3.8.11}$$

$$\boldsymbol{Q}^{(3)} = \frac{1}{k}\sum_{\kappa=1}^{k}(\boldsymbol{R}_\kappa - \boldsymbol{m}_\kappa\boldsymbol{m}_\kappa^{\mathsf{T}}) , \tag{3.8.12}$$

$$\boldsymbol{Q}^{(4)} = \frac{1}{k(k-1)}\sum_{\kappa=2}^{k}\sum_{\lambda=1}^{\kappa-1}(\boldsymbol{m}_\kappa - \boldsymbol{m}_\lambda)\,(\boldsymbol{m}_\kappa - \boldsymbol{m}_\lambda)^{\mathsf{T}} , \tag{3.8.13}$$

$$\boldsymbol{Q}^{(2)} = \boldsymbol{Q}^{(3)} + \boldsymbol{Q}^{(4)} . \tag{3.8.14}$$

Dabei ist $\boldsymbol{Q}^{(1)}$ die (symmetrische) **Kovarianzmatrix** der Muster in der Stichprobe. Alle obigen Matrizen haben die Größe $M \times M$, wenn M die Zahl der Abtastwerte des Musters $\boldsymbol{f}$ ist.

Die Matrizen lassen sich in der Form

$$\boldsymbol{Q} = \boldsymbol{V}\,\boldsymbol{V}^{\mathsf{T}} . \tag{3.8.15}$$

faktorisieren, was im Hinblick auf die Berechnung der Eigenvektoren u. U. zweckmäßig ist, wie weiter unten noch ausgeführt wird. Die Faktoren der verschiedenen Kernmatrizen sind

$$\boldsymbol{V}^{(1)\,M\times N} = \frac{1}{\sqrt{N}}\left({}^1\boldsymbol{f} - \boldsymbol{m},\ \ldots,\ {}^N\boldsymbol{f} - \boldsymbol{m}\right) , \tag{3.8.16}$$

$$\boldsymbol{V}^{(2)} = \left(\boldsymbol{V}^{(3)} \mid \boldsymbol{V}^{(4)}\right)\left(\boldsymbol{V}^{(3)} \mid \boldsymbol{V}^{(4)}\right)^{\mathsf{T}} , \tag{3.8.17}$$

$$\boldsymbol{V}^{(3)} = (\boldsymbol{V}_1,\ldots,\boldsymbol{V}_\kappa,\ldots,\boldsymbol{V}_k) , \tag{3.8.18}$$

$$\boldsymbol{V}_\kappa = \frac{1}{\sqrt{N_\kappa}}\left({}^1\boldsymbol{f}_\kappa - \boldsymbol{m}_\kappa,\ldots,{}^{N_\kappa}\boldsymbol{f}_\kappa - \boldsymbol{m}_\kappa\right) , \tag{3.8.19}$$

$$\boldsymbol{V}^{(4)} = \frac{1}{\sqrt{k}}\left(\boldsymbol{m}_1 - \overline{\boldsymbol{m}},\ldots,\boldsymbol{m}_k - \overline{\boldsymbol{m}}\right) , \quad \overline{\boldsymbol{m}} = \frac{1}{k}\sum_{\kappa=1}^{k}\boldsymbol{m}_\kappa . \tag{3.8.20}$$

$$\boldsymbol{Q}^{(l)\,M\times M} = \boldsymbol{V}^{(l)\,M\times N}\boldsymbol{V}^{(l)^{\mathsf{T}}N\times M} , \quad l = 1,2,3,4 . \tag{3.8.21}$$

Beweis zu Satz 3.8: Grundlage des Beweises ist die bekannte Tatsache, dass eine *quadratische Form* $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{Q}\boldsymbol{x}$, in der $\boldsymbol{x}$ ein beliebiger Vektor und $\boldsymbol{Q}$ eine positiv definite symmetrische Matrix ist, dann ihren Maximalwert (bzw. Minimalwert) annimmt, wenn $\boldsymbol{x}$ der *Eigenvektor* ist, der zum größten (bzw. kleinsten) Eigenwert von $\boldsymbol{Q}$ gehört. Der zweitgrößte (bzw. zweitkleinste) Wert wird angenommen, wenn man den zum zweitgrößten (bzw. zweitkleinsten) Eigenwert gehörigen Eigenvektor wählt, usw. Wie in (3.2.1) eingeführt, betrachten wir nur normierte

Vektoren. Es bleibt also zu zeigen, dass s_1, s_2, s_3 die Ausnutzung dieser Eigenschaft gestatten. Dieses wird hier am Beispiel von s_2 gezeigt. Man kann das in Satz 3.8 enthaltene Ergebnis auch für kontinuierliche Funktionen $f(x)$ ableiten. Für einen Vektor $\boldsymbol{f}$ von Abtastwerten erhält man durch Einsetzen von (3.2.2) in (3.8.2)

$$\begin{aligned} s_2 &= \frac{2}{k(k-1)} \sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-1} \frac{1}{N_\kappa N_\lambda} \sum_{i=1}^{N_\kappa} \sum_{j=1}^{N_\lambda} \\ &\quad \left({}^i\boldsymbol{f}_\kappa^\mathsf{T} \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi}\, {}^i\boldsymbol{f}_\kappa + {}^j\boldsymbol{f}_\lambda^\mathsf{T} \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi}\, {}^j\boldsymbol{f}_\lambda - {}^j\boldsymbol{f}_\lambda^\mathsf{T} \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi}\, {}^i\boldsymbol{f}_\kappa - {}^i\boldsymbol{f}_\kappa^\mathsf{T} \boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi}\, {}^j\boldsymbol{f}_\lambda \right) . \end{aligned}$$

Mit der für symmetrische Matrizen $\boldsymbol{Q}$ gültigen Beziehung

$$\boldsymbol{x}^\mathsf{T} \boldsymbol{Q} \boldsymbol{x} = \mathrm{Sp}\left(\boldsymbol{Q} \boldsymbol{x} \boldsymbol{x}^\mathsf{T}\right) , \tag{3.8.22}$$

wobei $\mathrm{Sp}\,(\boldsymbol{A})$ die Spur der Matrix $\boldsymbol{A}$ ist, also die Summe der Hauptdiagonalelemente, gilt

$$\begin{aligned} s_2 &= \frac{2}{k(k-1)} \sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-1} \frac{1}{N_\kappa N_\lambda} \sum_{i=1}^{N_\kappa} \sum_{j=1}^{N_\lambda} \\ &\quad \mathrm{Sp}\,\left(\boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi} \left({}^i\boldsymbol{f}_\kappa\, {}^i\boldsymbol{f}_\kappa^\mathsf{T} + {}^j\boldsymbol{f}_\lambda\, {}^j\boldsymbol{f}_\lambda^\mathsf{T} - {}^j\boldsymbol{f}_\lambda\, {}^i\boldsymbol{f}_\kappa^\mathsf{T} - {}^i\boldsymbol{f}_\kappa\, {}^j\boldsymbol{f}_\lambda^\mathsf{T} \right) \right) . \end{aligned}$$

Berücksichtigt man, dass mit (3.8.11)

$$\begin{aligned} \frac{1}{N_\kappa N_\lambda} \sum_{i=1}^{N_\kappa} \sum_{j=1}^{N_\lambda} {}^i\boldsymbol{f}_\kappa\, {}^i\boldsymbol{f}_\kappa^\mathsf{T} &= \frac{1}{N_\lambda} \sum_{j=1}^{N_\lambda} \boldsymbol{R}_\kappa = \boldsymbol{R}_\kappa , \\ \frac{1}{N_\kappa N_\lambda} \sum_{i=1}^{N_\kappa} \sum_{j=1}^{N_\lambda} {}^i\boldsymbol{f}_\kappa\, {}^j\boldsymbol{f}_\lambda^\mathsf{T} &= \boldsymbol{m}_\kappa \boldsymbol{m}_\lambda^\mathsf{T} \end{aligned} \tag{3.8.23}$$

ist, so vereinfacht sich s_2 zu

$$s_2 = \frac{2}{k(k-1)} \sum_{\kappa} \sum_{\lambda} \mathrm{Sp}\,\left(\boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi} (\boldsymbol{R}_\kappa + \boldsymbol{R}_\lambda - \boldsymbol{m}_\kappa \boldsymbol{m}_\lambda^\mathsf{T} - \boldsymbol{m}_\lambda \boldsymbol{m}_\kappa^\mathsf{T}) \right) .$$

Für die Matrix $\boldsymbol{\Phi}$ in (3.8.7) mit n Zeilen und M Spalten sowie für eine $M \times M$ Matrix $\boldsymbol{Q}$ gilt

$$\mathrm{Sp}\,(\boldsymbol{\Phi}^\mathsf{T} \boldsymbol{\Phi} \boldsymbol{Q}) = \sum_{\nu=1}^{n} \boldsymbol{\varphi}_\nu^\mathsf{T} \boldsymbol{Q} \boldsymbol{\varphi}_\nu , \tag{3.8.24}$$

was beispielsweise durch Vergleich der Summen auf der linken und rechten Seite dieser Beziehung leicht zu zeigen ist. Damit vereinfacht sich s_2 weiter zu

$$s_2 = \sum_{\nu=1}^{n} \boldsymbol{\varphi}_\nu^\mathsf{T} \left(\frac{2}{k(k-1)} \sum_{\kappa} \sum_{\lambda} (\boldsymbol{R}_\kappa + \boldsymbol{R}_\lambda - \boldsymbol{m}_\kappa \boldsymbol{m}_\lambda^\mathsf{T} - \boldsymbol{m}_\lambda \boldsymbol{m}_\kappa^\mathsf{T}) \right) \boldsymbol{\varphi}_\nu .$$

Eine weitere Vereinfachung erhält man mit der Beziehung

$$\sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-1} (\boldsymbol{R}_\lambda + \boldsymbol{R}_\kappa) = (k-1) \sum_{\kappa=1}^{k} \boldsymbol{R}_\kappa . \tag{3.8.25}$$

Damit und mit der in (3.8.10) definierten Matrix $\boldsymbol{Q}^{(2)}$ ist

$$s_2 = 2\sum_{\nu=1}^{n} \boldsymbol{\varphi}_\nu^{\mathsf{T}} \boldsymbol{Q}^{(2)} \boldsymbol{\varphi}_\nu \ . \tag{3.8.26}$$

Man sieht, dass $\boldsymbol{Q}^{(2)}$ symmetrisch ist, und wegen der Definition von s_2 als Abstandsquadrat ist sie auch positiv definit, d. h. es gilt $\boldsymbol{x}^{\mathsf{T}} \boldsymbol{Q}^{(2)} \boldsymbol{x} > 0, \forall \boldsymbol{x} \neq 0$. Damit ist gezeigt, dass s_2 aus n Summanden besteht, von denen jeder eine quadratische Form mit positiv definitem symmetrischem Kern ist. Aufgrund der erwähnten Eigenschaften solcher Formen wird s_2 dann maximiert, wenn man als Vektoren $\boldsymbol{\varphi}_\nu,\ \nu = 1, \ldots, n$ die n Eigenvektoren von $\boldsymbol{Q}^{(2)}$ wählt, die zu den n größten Eigenwerten gehören. Das ist aber gerade die Aussage von Satz 3.8. Eine ganz analoge Rechnung lässt sich für s_1 und s_3 durchführen. Damit ist der Satz 3.8 bewiesen.

Für einen Vektor $\boldsymbol{f}$ mit M Abtastwerten, wie in Abschnitt 2.1.1 ausgeführt, wird bei der Merkmalsgewinnung gemäß (3.2.2) stets $n \leq M$ sein, d. h. man hat *weniger* Merkmale c_ν als Abtastwerte f_i. Ist die Zahl N der Stichprobenelemente ${}^j\boldsymbol{f}$ *größer* als die Zahl M der Abtastwerte, also $N > M$, und sind die Muster ${}^j\boldsymbol{f},\ j = 1, \ldots, N$ *linear unabhängig*, so hat die $M \times M$ Matrix $\boldsymbol{Q}^{(2)}$ den Rang M mit Wahrscheinlichkeit Eins. Es ist bekannt, dass $\boldsymbol{Q}^{(2)}$ dann genau M verschiedene positive reelle Eigenwerte und M verschiedene orthogonale Eigenvektoren $\boldsymbol{\varphi}_\nu,\ \nu = 1, \ldots, M$ hat.

Wenn man ein Bild der Größe $M_x \times M_y$ betrachtet, so wird daraus durch Aneinanderreihen der Zeilen (oder der Spalten) ein Vektor der Länge $M = M_x \times M_y$ gebildet und davon eine Kernmatrix $\boldsymbol{Q}$ der Größe $M^2 = (M_x \times M_y)^2$ berechnet. Wenn man z. B. ein Bild der (moderaten) Größe 128×128 hat, ist $\boldsymbol{Q}$ bereits eine Matrix der Größe 16.384×16.384. Bei diesen Größenordnungen wird man u. U. *nicht mehr* die Bedingung $N > M$ einhalten können und numerische Probleme mit der großen Matrix bekommen. In diesem Falle hat man natürlich auch nicht mehr M positive reelle Eigenwerte, sondern nur noch $N < M$. Statt der Eigenwerte und –vektoren der sehr großen Matrix $\boldsymbol{Q}^{M\times M}$ kann man dann die einer kleineren Matrix $\overline{\boldsymbol{Q}}^{N\times N}$ berechnen. Zu dem Zweck wird $\boldsymbol{Q}^{M\times M}$ wie in (3.8.15) faktorisiert in

$$\boldsymbol{Q}^{M\times M} = \boldsymbol{V}^{M\times N} \boldsymbol{V}^{\mathsf{T}N\times M} \tag{3.8.27}$$

und mit den Faktoren eine neue Matrix $\overline{\boldsymbol{Q}}^{N\times N}$ definiert durch

$$\overline{\boldsymbol{Q}}^{N\times N} = \boldsymbol{V}^{\mathsf{T}N\times M} \boldsymbol{V}^{M\times N} \ . \tag{3.8.28}$$

Die Eigenwerte und –vektoren der neuen Matrix ergeben sich analog zu (3.8.6) aus

$$\overline{\boldsymbol{Q}}\,\overline{\boldsymbol{\varphi}}_\nu = \overline{\lambda}_\nu \overline{\boldsymbol{\varphi}}_\nu \ . \tag{3.8.29}$$

Setzt man oben (3.8.28) ein und multipliziert von links her mit $\boldsymbol{V}$, so erhält man

$$\begin{aligned} \boldsymbol{V}^{\mathsf{T}N\times M} \boldsymbol{V}^{M\times N} \overline{\boldsymbol{\varphi}}_\nu &= \overline{\lambda}_\nu \overline{\boldsymbol{\varphi}}_\nu \ , \\ \boldsymbol{V}\boldsymbol{V}^{\mathsf{T}} (\boldsymbol{V}\overline{\boldsymbol{\varphi}}_\nu) &= \overline{\lambda}_\nu (\boldsymbol{V}\overline{\boldsymbol{\varphi}}_\nu) \ , \\ \boldsymbol{Q} (\boldsymbol{V}\overline{\boldsymbol{\varphi}}_\nu) &= \overline{\lambda}_\nu (\boldsymbol{V}\overline{\boldsymbol{\varphi}}_\nu) \ , \end{aligned} \tag{3.8.30}$$

$$\boldsymbol{\varphi}_\nu = \frac{1}{|\boldsymbol{V}\overline{\boldsymbol{\varphi}}_\nu|} \boldsymbol{V}\overline{\boldsymbol{\varphi}}_\nu \ , \quad \lambda_\nu = \overline{\lambda}_\nu \ . \tag{3.8.31}$$

Man sieht, dass Eigenwerte der Matrix $\boldsymbol{Q}^{M\times M}$ auch Eigenwerte der Matrix $\overline{\boldsymbol{Q}}^{N\times N}$ sind. Ihre Eigenvektoren hängen über (3.8.31) zusammen. Man kann also wahlweise mit der einen oder anderen Kernmatrix rechnen, je nach dem, welche kleiner ist.

Von den berechneten Eigenvektoren werden $n < M$ Eigenvektoren gemäß Satz 3.8 zur Merkmalsgewinnung verwendet. Die Eigenvektoren seien wie in (3.2.1) auf den Betrag Eins normiert und die Eigenwerte so geordnet, dass $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$ ist. Der größte Wert von s_2 ist mit (3.8.26), (3.8.6)

$$s_{2,\max} = 2 \sum_{\nu=1}^{n} \lambda_\nu^{(2)} \ . \tag{3.8.32}$$

Eine entsprechende Gleichung erhält man für s_1, während der kleinste Wert für s_3 sich zu

$$s_{3,\min} = 2 \sum_{\nu=M-n+1}^{M} \lambda_\nu^{(3)} \tag{3.8.33}$$

ergibt. Zur Berechnung der Matrizen $\boldsymbol{Q}^{(l)}$ ist eine Stichprobe ω von Mustern erforderlich, deren Klassenzugehörigkeit für $\boldsymbol{Q}^{(2)}$ und $\boldsymbol{Q}^{(3)}$ *bekannt* sein muss, für $\boldsymbol{Q}^{(1)}$ dagegen *nicht*. Wegen der Abhängigkeit von $\boldsymbol{Q}^{(l)}$ und damit auch von $\varphi_\nu^{(l)}$ von der Stichprobe ω werden diese orthonormalen Entwicklungen, wie anfangs erwähnt, auch als problemabhängig bezeichnet.

Hauptachsentransformation

Man bezeichnet die durch $s_1, \boldsymbol{Q}^{(1)}$ definierte lineare Transformation auch als diskrete KARHUNEN–LOÈVE-**Transformation** (KLT) oder **Hauptachsentransformation** (HAT). Außer der Maximierung von s_1 hat sie noch die Eigenschaften, dass die Entwicklungskoeffizienten c_ν *unkorreliert* sind und dass der mittlere quadratische Approximationsfehler *minimiert* wird. Oft wird, abweichend von (3.2.2), S. 166, diese Transformation auch durch

$${}^j\boldsymbol{c}' = \boldsymbol{\Phi}({}^j\boldsymbol{f} - \boldsymbol{m}) = {}^j\boldsymbol{c} - \boldsymbol{\mu} \tag{3.8.34}$$

definiert, wobei $\boldsymbol{m}$ der in (3.8.9) definierte Mittelwertsvektor der Stichprobe ist. Offensichtlich wird mit (3.8.34) die Punktmenge $\{{}^j\boldsymbol{c} \mid j = 1, \ldots, N\}$ lediglich um $\boldsymbol{\mu}$ verschoben, aber die relative Lage der Punkte zueinander bleibt unverändert. Zumindest für die Klassifikation sind (3.2.2) und (3.8.34) also äquivalent. Eine anschauliche Vorstellung von der Wirkung der Hauptachsentransformation gibt Bild 3.8.1.

Wegen der besonderen Bedeutung der Hauptachsentransformation in der Mustererkennung werden ihre Eigenschaften am Beispiel der Stichprobe COIL-20 (COIL: Columbia Object Image Library) etwas genauer demonstriert. Die Stichprobe wird verwendet, da sie weit verbreitet ist und zudem relativ klein ist, sodass Berechnungen leicht durchführbar sind. Sie zeigt, wie schon in Abschnitt 1.9 erwähnt, Bilder von 20 Objekten in 72 Drehlagen, also insgesamt 1440 Bilder, in der Auflösung 128×128. Beispiele sind in Bild 3.8.2 gezeigt.

Die Bilder in Bild 3.8.3 zeigen einige Eigenschaften der Eigenwerte, die zur Reduktion des Rechenaufwandes auf Bildern der Größe 64×64 berechnet wurden (mit dem Programm „eigs“ aus Matlab). Man sieht, dass die Eigenwerte mit zunehmender Ordnung sehr rasch abnehmen. Wegen (3.8.32) hängt der Wert des Abstandskriteriums s_2 von der Summe der verwendeten Eigenwerte ab. Aus Bild 3.8.3 geht hervor, dass diese Summe zunächst sehr rasch wächst, ab etwa $M = 60$ Eigenwerten bereits rund 90% der maximal mglichen Summe erreicht und dann nur noch sehr flach ansteigt. Daraus folgt der Schluss, dass nur relativ wenige Eigenvektoren und damit relativ wenige Merkmale für die Klassifikation ausreichen sollten.

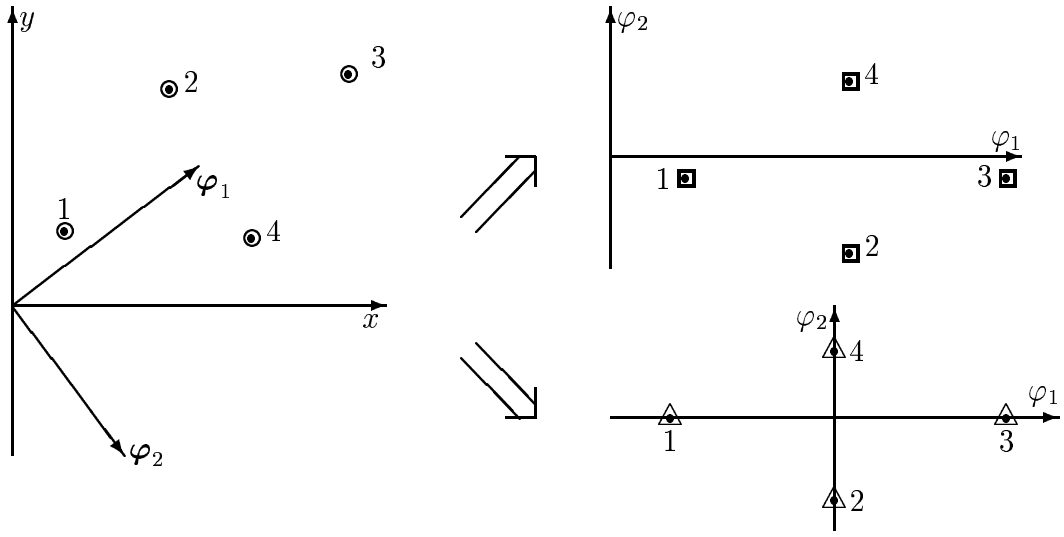


Bild 3.8.1: Hauptachsentransformation einer Punktmenge, rechts oben mit (3.2.2), rechts unten mit (3.8.34)

Bild 3.8.2: Die 20 Objekte der COIL-20 Stichprobe in unterschiedlichen Drehlagen

Die Bilder Bild 3.8.4 – Bild 3.8.5 zeigen Eigenvektoren niedriger und hoher Ordnung. Man sieht, dass die ersten Eigenwerte eher niederfrequente Anteile enthalten, die höherer Ordnung zunehmend hochfrequente.

Schließlich zeigt Bild 3.8.6 einige Beispiele für die Approximation der gegebenen Bilder mit unterschiedlich vielen Eigenvektoren. Eine erkennbare Darstellung wird bereits mit 40 – 60 Eigenvektoren erreicht, alle Details sind auch bei 800 noch nicht vollständig approximiert. Da der Stichprobenumfang $N = 1440$ Bilder beträgt, gibt es maximal 1440 von Null verschiedene Eigenwerte, d.h. die perfekte Approximation erfordert maximal 1440 Eigenvektoren.

Problemabhängige Entwicklungen in verschiedenen Modifikationen, darunter die Hauptachsentransformation, wurden für die Mustererkennung schon relativ früh vorgeschlagen und werden seitdem sowohl theoretisch als auch experimentell immer wieder aufgegriffen. Ein im-

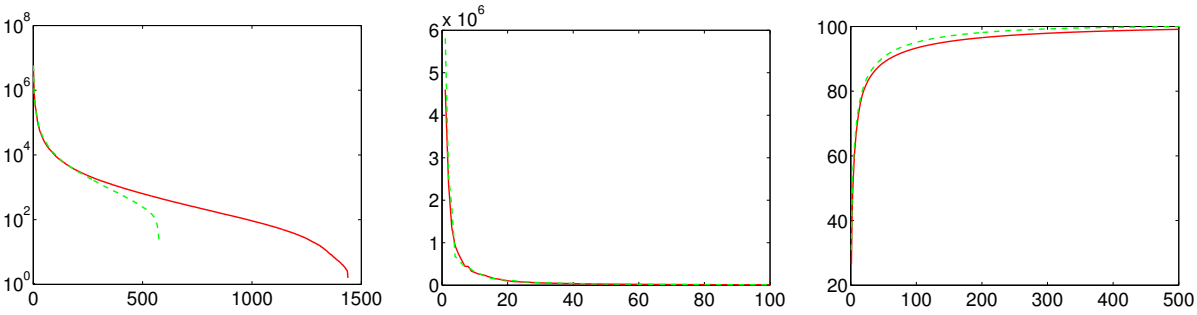


Bild 3.8.3: Die Eigenwerte der COIL-20 Stichprobe; links im logarithmischen Massstab alle Eigenwerte der vollen Stichprobe von 20 Objekten in 72 Drehlagen und der ersten 8 Objekte in 72 Drehlagen; in der Mitte im linearen Massstab die ersten 100 Eigenwerte fuer die 20 und 8 Objekte; rechts in % für die ersten 500 Eigenwerte das Verhältnis der Summe der ersten $i = 1, 2, \ldots, 500$ Eigenwerte zur Summe aller Eigenwerte; jeweils durchgezogene Linie für die Bilder von 20 Objekten, gestrichelte für die von 8 Objekten

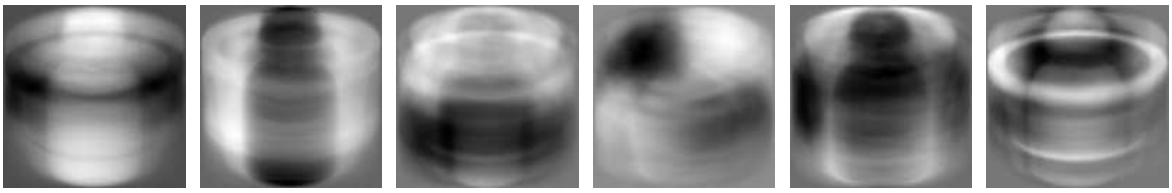


Bild 3.8.4: Die ersten 6 Eigenvektoren der COIL-20 Stichprobe

mer wieder bestätigtes Ergebnis – das auch wegen Bild 3.8.6 plausibel ist – ist, dass bei mehr als $n = 20$ bis 30 Merkmalen die Fehlerrate bei der Klassifikation kaum noch abnimmt und dass die problemabhängigen Entwicklungen bei gleicher Merkmalszahl kleinere Fehlerraten ergeben als die problemunabhängigen. Die verschiedenen Formen der problemabhängigen Entwicklungen bieten Vorteile vor allem bei kleiner Anzahl von Merkmalen. Ein wichtiger Vorteil der genannten Entwicklungen ist – neben der Extraktion von nur wenigen aber wichtigen Merkmalen – dass die Merkmale gemäß (3.2.2) zumindest näherungsweise normalverteilt sind; dieses wird auch durch den zentralen Grenzwertsatz der Statistik nahegelegt. Wie in Kapitel 4 ausgeführt wird, sind Kenntnisse über die statistischen Eigenschaften der Merkmale Voraussetzung für den Einsatz statistischer Klassifikatoren.

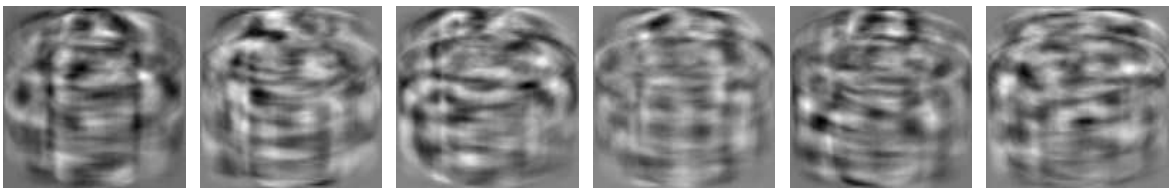
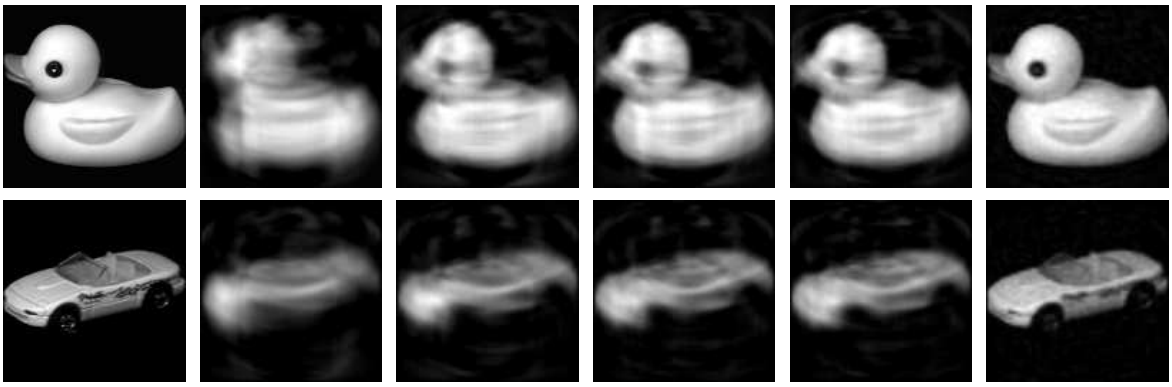


Bild 3.8.5: Die Eigenvektoren Nr. 60 – 65 der COIL-20 Stichprobe



Original $n = 20$ $n = 40$ $n = 60$ $n = 80$ $n = 800$

Bild 3.8.6: Die Approximation des 1. und 19. Bildes aus Bild 3.8.2 mit unterschiedlicher Zahl n von Eigenvektoren

Kombinierte Kriterien

Die Vergrößerung des Interklassenabstandes (3.8.2) hat nur dann einen offensichtlichen Vorteil, wenn gleichzeitig der Intraklassenabstand *konstant* bleibt, bzw. sogar verkleinert wird. Dieses kann durch eine Kombination von Kriterien erreicht werden. Mögliche Kombinationen von jeweils zwei der obigen Abstände sind die Kriterien

$$s_{5,1} = s_2 + \vartheta s_3 \ , \tag{3.8.35}$$

$$s_{5,2} = s_4 + \vartheta s_3 \ , \tag{3.8.36}$$

$$s_{5,3} = \frac{s_2}{s_3} \ . \tag{3.8.37}$$

wobei ϑ ein LAGRANGE-*Multiplikator* ist. Das Kriterium $s_{5,3}$ ist das FISHER-**Kriterium**. Die Kernmatrizen sind

$$\boldsymbol{Q}^{(5,1)} = \boldsymbol{Q}^{(2)} + \vartheta \boldsymbol{Q}^{(3)} \ , \quad \boldsymbol{Q}^{(5,2)} = \boldsymbol{Q}^{(4)} + \vartheta' \boldsymbol{Q}^{(3)} \ . \tag{3.8.38}$$

Das Kriterium $s_{5,1}$ in (3.8.35) lässt sich so interpretieren, dass eine Transformationsmatrix $\boldsymbol{\Phi}$ gesucht wird, welche s_2 maximiert unter der Nebenbedingung $s_3 = \text{const}$. Dieses ist intuitiv eine vernünftige Forderung, da ein großer Wert von s_2, also ein großer Interklassenabstand, dann sinnlos ist, wenn gleichzeitig s_3, also der Intraklassenabstand, groß wird. Es sei angemerkt, dass bei Fehlen dieser Nebenbedingung trotzdem die Triviallösung $s_2 \to \infty$ ausgeschlossen ist, da die $\boldsymbol{\varphi}_\nu$ als normiert vorausgesetzt werden. Mit der Nebenbedingung $s_3 = \text{const}$ lässt sich zwar die Kernmatrix in (3.8.38) ableiten, jedoch ist es nicht möglich, den Wert des LAGRANGE-Multiplikators ϑ geschlossen zu berechnen. Man kann ihn allerdings näherungsweise ermitteln, indem man für verschiedene Werte von ϑ die zugehörige Transformationsmatrix $\boldsymbol{\Phi}^{(5)}$ gemäß Satz 3.8 bestimmt, ein Klassifikationssystem realisiert und die damit erreichbare Fehlerrate schätzt. Derjenige Wert von ϑ, der die kleinste Fehlerrate ergibt, wird für die Merkmalsgewinnung verwendet.

Die obige Vorgehensweise hat engen Zusammenhang zur **Diskriminanzanalyse**. Diese optimiert das Kriterium

$$s_{5,4} = \mathrm{Sp}(\boldsymbol{Q}_1^{-1}\boldsymbol{Q}_5) \ , \tag{3.8.39}$$

$$\begin{aligned}\boldsymbol{Q}_1 &= E\{(\boldsymbol{f}-E\{\boldsymbol{f}\})(\boldsymbol{f}-E\{\boldsymbol{f}\})^\mathsf{T}\}\,,\\ \boldsymbol{Q}_5 &= \sum_{\kappa=1}^{k} p_\kappa (E_\kappa\{\boldsymbol{f}\}-E\{\boldsymbol{f}\})(E_\kappa\{\boldsymbol{f}\}-E\{\boldsymbol{f}\})^\mathsf{T}\,.\end{aligned}$$

Dabei ist $E_\kappa\{\cdot\}$ der durch Ω_κ bedingte **Erwartungswert**. Offensichtlich ist $\boldsymbol{Q}^{(1)}$ in (3.8.8) ein Schätzwert für $\boldsymbol{Q}_1$ und $\boldsymbol{Q}_5$ ein ungefähres Maß für die Abstände der Klassen voneinander. Mit einer weiteren Matrix

$$\boldsymbol{Q}_3 = \sum_{\kappa=1}^{k} p_\kappa E_\kappa\{(\boldsymbol{f}-E_\kappa\{\boldsymbol{f}\})(\boldsymbol{f}-E_\kappa\{\boldsymbol{f}\})^\mathsf{T}\}\,, \tag{3.8.40}$$

für die $\boldsymbol{Q}^{(3)}$ in (3.8.12) ein Schätzwert mit $p_\kappa = \frac{1}{k}$ ist, gilt

$$\boldsymbol{Q}_1 = \boldsymbol{Q}_3 + \boldsymbol{Q}_5 \tag{3.8.41}$$

Damit lässt sich das Kriterium $s_{5,4}$ ähnlich deuten wie $s_{5,3}$. Ein großer Wert von $s_{5,4}$ wird erreicht, wenn Muster dicht beisammen liegen, denn dann ist $\mathrm{Sp}(\boldsymbol{Q}_1)$ klein, und wenn die Klassen gut getrennt sind, denn dann ist $\mathrm{Sp}(\boldsymbol{Q}_5)$ groß und damit auch $\mathrm{Sp}(\boldsymbol{Q}_1^{-1}\boldsymbol{Q}_5)$ groß. s_5 angeben lässt, dessen Interpretation wie die von s_5' ist. Gesucht wird wieder die Transformationsmatrix, die $s_{5,4}$ maximiert. Analog zu Satz 3.8 sind die Zeilen der Transformationsmatrix $\boldsymbol{\Phi}$ die zu den n größten Eigenwerten gehörigen Eigenvektoren von $\boldsymbol{Q}_1^{-1}\boldsymbol{Q}_5$. Vernachlässigt man $\boldsymbol{Q}_1^{-1}$, setzt $k=2$ und betrachtet nur skalare „Muster“ f, so geht (3.8.39) in (2.2.15), S. 83, über.

3.8.3 Nichtlineare (kernbasierte) Hauptachsentransformation

Generelle Möglichkeiten

In das Optimierungskriterium s_1 in (3.8.1) gehen nur die Komponenten des Merkmalsvektors $\boldsymbol{c}$ ein, und damit ergibt sich in (3.2.2) eine Transformation, die *linear* in den Komponenten des Musters $\boldsymbol{f}$ ist. Eine naheliegende Verallgemeinerung besteht darin, ein „erweitertes“ Muster zu betrachten, das auch Produktterme in den Komponenten f_j enthält. Dieses Prinzip wird auch beim Polynomklassifikator in Abschnitt 4.4 oder bei den Support Vektor Maschinen in Abschnitt 4.3 zur Verallgemeinerung der Trennfunktionen verwendet. Statt des Musters $\boldsymbol{f}$ wird also ein im Prinzip beliebig transformiertes betrachtet. Speziell für Polynome vom Grad p in den Komponenten ergibt sich

$$\boldsymbol{\phi}(\boldsymbol{f}) = \left(1, f_1, \ldots, f_M, f_1f_1, f_2f_1, \ldots, f_Mf_M, \ldots, f_{j_1}f_{j_2}\ldots f_{j_p}, \ldots\right)^\mathsf{T}\,. \tag{3.8.42}$$

An den Ausführungen zu problemabhängigen Reihenentwicklungen in Abschnitt 3.8.2, an Satz 3.8 und an den Ergebnissen in (3.8.7) – (3.8.12) ändert sich offenbar nichts, wenn man dort das Muster $\boldsymbol{f}$ durch sein transformiertes $\boldsymbol{\phi}(\boldsymbol{f})$ ersetzt.

Allerdings wird die Komplexität der Rechnungen drastisch erhöht, da ein Polynom vom Grade p mit M Variablen

$$n_a = \binom{M+p}{p} = \frac{(M+p)!}{M!\,p!} \tag{3.8.43}$$

Terme und damit auch Koeffizienten hat. Die Zahl der *Monome* $f_{j_1}f_{j_2}\ldots f_{j_p}$ ist

$$n_m = \frac{(M+p-1)!}{(M-1)!\,p!}\,. \tag{3.8.44}$$

Schon für ein Muster mit der mäßigen Auflösung von $M = 16 \times 16$ Abtastwerten und ein Polynom vom Grade $p = 3$ ergibt sich $n_a \approx 2,8 \cdot 10^6$. Die Matrix Q_1 hätte also die Größe $(2,8 \cdot 10^6)^2$. Diese Vorgehensweise ist damit, wenn überhaupt, nur eingeschränkt nutzbar. Eine mögliche Einschränkung besteht darin, nicht ein vollständiges Polynom zu verwenden, sondern zu jeder Komponente f_j nur Produktterme mit Komponenten aus einer kleinen heuristisch gewählten Nachbarschaft zu bilden.

Wenn man allerdings das Muster $\boldsymbol{f}$ zunächst in einen Merkmalsvektor $\boldsymbol{c}$ mit deutlich weniger Komponenten transformiert, kann man dann diese Vorgehensweise auf den Merkmalsvektor anwenden. Statt einer Transformation $\boldsymbol{\phi}(\boldsymbol{f})$ wird also $\boldsymbol{\phi}(\boldsymbol{c})$ betrachtet. Wird zu einem Merkmalsvektor mit $n = 30$ Komponenten ein Polynom vom Grade $p = 3$ gebildet, so hat dieses „nur“ $n_a \approx 5500$ Terme.

Eine weitere Möglichkeit bietet die Beobachtung, dass sich Skalarprodukte von transformierten Vektoren, also Produkte der Form $\boldsymbol{\phi}(\boldsymbol{f})^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{f})$, auf die Berechnung von Skalarprodukten der untransformierten Vektoren, also Produkte der Form $\boldsymbol{f}^\mathsf{T}\boldsymbol{f}$, zurückführen lassen. Damit reduziert sich die Rechenkomplexität für die Skalarprodukte von $\mathcal{O}(n_a)$ auf $\mathcal{O}(n)$. Dieses ist immer dort nutzbar, wo sich Rechnungen auf Skalarprodukte zurückführen lassen, und das ist z. B. bei der Hauptachsentransformation oder den Support Vektor Maschinen der Fall.

Skalarprodukte von transformierten Merkmalsvektoren

Als einfaches Beispiel einer Transformation $\boldsymbol{\phi}$ in einen höherdimensionalen Raum betrachten wir einen Vektor $\boldsymbol{c}$ mit

$$\boldsymbol{c} = (c_1, c_2, c_3)^\mathsf{T} \tag{3.8.45}$$

und wählen einen transformierten Vektor $\boldsymbol{\phi}(\boldsymbol{c})$ zu

$$\boldsymbol{\phi}(\boldsymbol{c}) = \widetilde{\boldsymbol{c}} = (c_1^2, \sqrt{2}c_1c_2, c_2^2, \sqrt{2}c_2c_3, c_3^2, \sqrt{2}c_1c_3)^\mathsf{T} \ . \tag{3.8.46}$$

Die Skalarprodukte zweier Vektoren dieses Typs sind

$$\begin{aligned} {}^i\boldsymbol{c}^\mathsf{T}\,{}^j\boldsymbol{c} &= {}^ic_1\,{}^jc_1 + {}^ic_2\,{}^jc_2 + {}^ic_3\,{}^jc_3 \ , & (3.8.47)\\ {}^i\widetilde{\boldsymbol{c}}^\mathsf{T}\,{}^j\widetilde{\boldsymbol{c}} &= {}^ic_1^2\,{}^jc_1^2 + 2{}^ic_1\,{}^ic_2\,{}^jc_1\,{}^jc_2 + {}^ic_2^2\,{}^jc_2^2 + 2{}^ic_2\,{}^ic_3\,{}^jc_2\,{}^jc_3 \\ &\quad + {}^ic_3^2\,{}^jc_3^2 + 2{}^ic_1\,{}^ic_3\,{}^jc_1\,{}^jc_3 \ . & (3.8.48) \end{aligned}$$

Man sieht, dass sich mit einer geeigneten *Kernfunktion* K, in diesem Beispiel $K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = ({}^i\boldsymbol{c}^\mathsf{T}\,{}^j\boldsymbol{c})^2$, die Identität

$$K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \left({}^i\boldsymbol{c}^\mathsf{T}\,{}^j\boldsymbol{c}\right)^2 = {}^i\widetilde{\boldsymbol{c}}^\mathsf{T}\,{}^j\widetilde{\boldsymbol{c}} \tag{3.8.49}$$

ergibt. Das bedeutet, dass man statt der Berechnung der Abbildung $\boldsymbol{\phi}$ und des (langen) Skalarproduktes in einem hochdimensionalen Raum auch beim Ausgangsvektor im niederdimensionalen Raum bleiben und nur dessen Skalarprodukt mit der Kernfunktion K transformieren kann. Damit wird, wie gesagt, bei großen Merkmalsvektoren und dem Übergang auf hochdimensionale Räume die Komplexität der Rechnung enorm reduziert.

Im obigen Beispiel bekommt man *nicht* ein allgemeines Polynom in $\boldsymbol{c}$, da die linearen Terme fehlen. Wählt man jedoch

$$K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \left({}^i\boldsymbol{c}^\mathsf{T}\,{}^j\boldsymbol{c} + 1\right)^p \ , \tag{3.8.50}$$

so wird damit ein Polynom in den Komponenten des einen Merkmalsvektors vom Grade p generiert. Man überzeugt sich davon leicht an obigem Beispiel für $p = 2$. Zwei weitere Beispiele für Kernfunktionen sind

$$K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \exp\left[-\frac{1}{2}\frac{|{}^i\boldsymbol{c} - {}^j\boldsymbol{c}|^2}{\sigma^2}\right] , \tag{3.8.51}$$

$$K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \tanh\left[\gamma_1 {}^i\boldsymbol{c}^{\mathsf{T}} {}^j\boldsymbol{c} - \gamma_2\right] , \tag{3.8.52}$$

in denen der Polynomgrad sogar unendlich ist.

Die obige Vorgehensweise lässt sich offensichtlich immer dann anwenden, wenn zu einer nichtlinearen Abbildung $\boldsymbol{\phi}$ eine **Kernfunktion** K existiert, sodass

$$\boxed{K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \boldsymbol{\phi}({}^i\boldsymbol{c})^{\mathsf{T}} \boldsymbol{\phi}({}^j\boldsymbol{c})} \tag{3.8.53}$$

gilt. Zur Frage, zu welchen Kernen K es Abbildungen $\boldsymbol{\phi}$ gibt, gilt

Satz 3.9 *(Satz von* MERCER*) Eine Abbildung $\boldsymbol{\phi}$ mit Entwicklung*

$$K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \sum_\nu \boldsymbol{\phi}({}^i\boldsymbol{c})_\nu^{\mathsf{T}} \boldsymbol{\phi}({}^j\boldsymbol{c})_\nu \tag{3.8.54}$$

existiert genau dann, wenn für jede Funktion $g(\boldsymbol{c})$ mit

$$\int g(\boldsymbol{c})^2 \, \mathrm{d}\boldsymbol{c} < \infty \tag{3.8.55}$$

gilt

$$\int K(\boldsymbol{c}, \boldsymbol{c}') g(\boldsymbol{c}) g(\boldsymbol{c}') \, \mathrm{d}\boldsymbol{c} \, \mathrm{d}\boldsymbol{c}' \geq 0 \, . \tag{3.8.56}$$

Beweis*: s. z. B. [Courant und Hilbert, 1953], III, §5 und [Vapnik, 1995].*

Das Problem bei der Prüfung der Bedingung (3.8.56) für den Satz von MERCER ist, dass diese für *jedes* gemäß (3.8.55) quadratisch integrierbare $g(\boldsymbol{c})$ erfüllt sein muss. Für die Kerne in (3.8.49) – (3.8.52) trifft dieses zu.

Kernbasierte Hauptachsentransformation

Um das Ergebnis (3.8.53) für die Hauptachsentransformation zu nutzen, müssen die wesentlichen Rechnungen durch Skalarprodukte ausgedrückt werden. Während in (3.8.8) die Mittelwerte der Muster getrennt behandelt wurden, gehen wir im Folgenden von mittelwertfreien Mustern aus. Falls dieses nicht zutrifft, wird der Mittelwert vorher subtrahiert. Auch für mittelwertfreie Muster sind die transformierten Muster $\boldsymbol{\phi}(\boldsymbol{f})$ i. Allg. *nicht* mittelwertfrei; im Folgenden wird jedoch zunächst angenommen, dass auch diese mittelwertfrei sind. Es gelte also

$$E\{\boldsymbol{f}\} = E\{\boldsymbol{\phi}(\boldsymbol{f})\} = \mathbf{0} \, . \tag{3.8.57}$$

Damit ist die Kovarianzmatrix der Muster aus der Stichprobe mit (3.8.8) und (3.8.9)

$$\boldsymbol{Q}^{(1)} = \boldsymbol{R} = \frac{1}{N}\sum_{j=1}^{N} {}^{j}\boldsymbol{f}\,{}^{j}\boldsymbol{f}^{\mathsf{T}} \ . \tag{3.8.58}$$

Gemäß (3.8.6) werden die Eigenvektoren von $\boldsymbol{R}$ bestimmt aus der Gleichung

$$\begin{aligned} \lambda_\nu \boldsymbol{\varphi}_\nu &= \boldsymbol{R}\boldsymbol{\varphi}_\nu = \frac{1}{N}\sum_{j=1}^{N} {}^{j}\boldsymbol{f}\ \left({}^{j}\boldsymbol{f}^{\mathsf{T}}\boldsymbol{\varphi}_\nu\right) \\ &= \frac{1}{N}\sum_{j=1}^{N} \alpha_{\nu,j}\,{}^{j}\boldsymbol{f} \ . \end{aligned} \tag{3.8.59}$$

Aus der obigen Gleichung geht hervor, dass alle Eigenvektoren $\boldsymbol{\varphi}_\nu$ zu Eigenwerten $\lambda_\nu \neq 0$ in dem Raum liegen, der von den Mustern der Stichprobe aufgespannt wird.

Nach einer Transformation des Musters, z. B. wie in (3.8.42) oder speziell (4.4.5), S. 369, ergibt sich mit der Annahme (3.8.57) für die Kovarianzmatrix der transformierten Muster

$$\boldsymbol{S} = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\phi}({}^{j}\boldsymbol{f})\,\boldsymbol{\phi}({}^{j}\boldsymbol{f})^{\mathsf{T}} \ . \tag{3.8.60}$$

Gesucht sind nun die Eigenvektoren $\boldsymbol{\psi}$ der Matrix $\boldsymbol{S}$ durch Lösen der Gleichung

$$\vartheta_\nu \boldsymbol{\psi}_\nu = \boldsymbol{S}\,\boldsymbol{\psi}_\nu \ . \tag{3.8.61}$$

Aus der obigen Argumentation folgt, dass alle Eigenvektoren von $\boldsymbol{S}$ in dem von der transformierten Stichprobe $\omega_{\boldsymbol{\phi}} = \{\boldsymbol{\phi}({}^{1}\boldsymbol{f}), \ldots, \boldsymbol{\phi}({}^{N}\boldsymbol{f})\}$ aufgespannten Raum liegen. Daher gilt analog zu (3.8.59)

$$\boldsymbol{\psi}_\nu = \frac{1}{N}\sum_{i=1}^{N} \beta_{\nu,i}\boldsymbol{\phi}({}^{i}\boldsymbol{f}) \ , \tag{3.8.62}$$

und statt (3.8.61) kann man auch die Gleichungen

$$\vartheta_\nu \left(\boldsymbol{\phi}({}^{k}\boldsymbol{f})^{\mathsf{T}}\boldsymbol{\psi}_\nu\right) = \left(\boldsymbol{\phi}({}^{k}\boldsymbol{f})^{\mathsf{T}}\,\boldsymbol{S}\,\boldsymbol{\psi}_\nu\right) \ , \quad k = 1, \ldots, N \tag{3.8.63}$$

lösen. Durch Einsetzen von (3.8.62) und (3.8.60) in (3.8.63) ergibt sich

$$\begin{aligned} \vartheta_\nu \sum_{i=1}^{N} \beta_{\nu,i} \left(\boldsymbol{\phi}({}^{k}\boldsymbol{f})^{\mathsf{T}}\boldsymbol{\phi}({}^{i}\boldsymbol{f})\right) &= \frac{1}{N}\sum_{i=1}^{N} \beta_{\nu,i} \left(\boldsymbol{\phi}({}^{k}\boldsymbol{f})^{\mathsf{T}} \sum_{j=1}^{N} \boldsymbol{\phi}({}^{j}\boldsymbol{f}) \left(\boldsymbol{\phi}({}^{j}\boldsymbol{f})^{\mathsf{T}}\boldsymbol{\phi}({}^{i}\boldsymbol{f})\right)\right) \ , \\ & \qquad k = 1, \ldots, N \ . \end{aligned} \tag{3.8.64}$$

Wenn man die Koeffizienten $\beta_{\nu,i}$ im Vektor $\boldsymbol{\beta}_\nu$ zusammenfasst und eine N^2 Matrix $\boldsymbol{T}$ definiert mit

$$\boldsymbol{T} = \begin{pmatrix} \left(\boldsymbol{\phi}({}^{1}\boldsymbol{f})^{\mathsf{T}}\boldsymbol{\phi}({}^{1}\boldsymbol{f})\right) & \ldots & \left(\boldsymbol{\phi}({}^{1}\boldsymbol{f})^{\mathsf{T}}\boldsymbol{\phi}({}^{N}\boldsymbol{f})\right) \\ \vdots & \vdots & \vdots \\ \left(\boldsymbol{\phi}({}^{N}\boldsymbol{f})^{\mathsf{T}}\boldsymbol{\phi}({}^{1}\boldsymbol{f})\right) & \ldots & \left(\boldsymbol{\phi}({}^{N}\boldsymbol{f})^{\mathsf{T}}\boldsymbol{\phi}({}^{N}\boldsymbol{f})\right) \end{pmatrix} \ , \tag{3.8.65}$$

lässt sich (3.8.64) in der kompakten Form schreiben

$$\begin{aligned} N\vartheta_\nu \boldsymbol{T}\boldsymbol{\beta}_\nu &= \boldsymbol{T}^2\boldsymbol{\beta}_\nu \,, \\ N\vartheta_\nu \boldsymbol{\beta}_\nu &= \boldsymbol{T}\boldsymbol{\beta}_\nu \,. \end{aligned} \tag{3.8.66}$$

Damit hat man ein Eigenwertproblem zur Berechnung der Koeffizientenvektoren $\boldsymbol{\beta}_\nu$, bei dem die Matrix $\boldsymbol{T}$ nur die Berechnung von Skalarprodukten der transformierten Vektoren erfordert. Wegen (3.8.53) können diese Skalarprodukte berechnet werden, *ohne* die Vektoren tatsächlich zu transformieren; d. h. die Komplexität der Rechnung wird drastisch reduziert. Man erkennt, dass man dafür einen Preis zahlt, nämlich die Einführung der N^2 Matrix $\boldsymbol{T}$, wobei N der Stichprobenumfang ist. Diese Vorgehensweise ist also auf kleine Stichproben beschränkt.

Nach Berechnung der Koeffizientenvektoren $\boldsymbol{\beta}_\nu$ lassen sich die eigentlich gesuchten Entwicklungskoeffizienten

$$\widetilde{c}_\nu = \boldsymbol{\psi}_\nu^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{f}) \tag{3.8.67}$$

berechnen. Die Eigenvektoren $\boldsymbol{\psi}_\nu$ der Kovarianzmatrix (3.8.60) sollen auf den Betrag Eins normiert sein. Daraus ergibt sich mit (3.8.62), (3.8.65) und (3.8.66)

$$\begin{aligned} 1 &= \boldsymbol{\psi}_\nu^{\mathsf{T}} \boldsymbol{\psi}_\nu \,, \quad \nu = 1, \ldots, N \\ &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \beta_{\nu,i} \beta_{\nu,j} \left(\boldsymbol{\phi}({}^i\boldsymbol{f})^{\mathsf{T}} \boldsymbol{\phi}({}^j\boldsymbol{f}) \right) \\ &= \frac{1}{N^2} \boldsymbol{\beta}_\nu^{\mathsf{T}} \boldsymbol{T} \boldsymbol{\beta}_\nu \\ &= \frac{1}{N} \vartheta_\nu \left(\boldsymbol{\beta}_\nu^{\mathsf{T}} \boldsymbol{\beta}_\nu \right) \,. \end{aligned} \tag{3.8.68}$$

Das ist eine Normierungsbedingung für die Koeffizientenvektoren $\boldsymbol{\beta}_\nu$. Im Folgenden seien diese Vektoren gemäß (3.8.68) normiert. Mit (3.8.67) und (3.8.62) erhält man nun die gesuchten Entwicklungskoeffizienten der nichtlinearen Hauptachsentransformation eines Musters $\boldsymbol{f}$ aus

$$\boxed{\begin{aligned} \widetilde{c}_\nu &= \boldsymbol{\psi}_\nu^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{f}) = \frac{1}{N} \sum_{i=1}^{N} \beta_{\nu,i} \left(\boldsymbol{\phi}({}^i\boldsymbol{f})^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{f}) \right) \\ &= \frac{1}{N} \sum_{i=1}^{N} \beta_{\nu,i} \, K\left({}^i\boldsymbol{f}, \boldsymbol{f} \right) \,. \end{aligned}} \tag{3.8.69}$$

In (3.8.62) und damit auch in (3.8.69) wurde darauf verzichtet, den Faktor $1/N$ mit den Koeffizienten $\beta_{\nu,i}$ zusammenzufassen. Auch hier sind nur Skalarprodukte der transformierten Vektoren zu berechnen, was wegen (3.8.53) ohne Transformation der Muster möglich ist. Allerdings muss man wieder über die gesamte Stichprobe summieren.

Zusammengefasst besteht die nichtlinearen Hauptachsentransformation also aus folgenden Schritten. Es wird eine nichtlineare Transformation $\boldsymbol{\phi}$ gewählt, die dem Satz von MERCER genügt; Beispiele dafür sind (3.8.50) – (3.8.52). Man berechnet die Matrix $\boldsymbol{T}$ in (3.8.65) und deren Eigenvektoren, die zu positiven Eigenwerten gehören. Die Eigenvektoren werden gemäß (3.8.68) normiert. Schliesslich erhält man die Entwicklungskoeffizienten aus (3.8.69).

Es sollte beachtet werden, dass man also die Entwicklungskoeffizienten $\widetilde{c}_\nu$ in (3.8.67) der KL–Transformation auf zwei Arten berechnen kann. Zum einen über die $n_a \times n_a$ Matrix $\boldsymbol{S}$

in (3.8.60) und Lösung von (3.8.61); dabei ist n_a die Zahl der Komponenten von $\boldsymbol{\phi}(\boldsymbol{f})$. Zum anderen über die $N \times N$ Matrix $\boldsymbol{T}$ in (3.8.65), Lösung von (3.8.66) und Verwendung von (3.8.69); dabei ist N der Stichprobenumfang. Da $\boldsymbol{\phi}$ beliebig sein kann, gilt das natürlich auch für die KL–Transformation der untransformierten Muster $\boldsymbol{f}$ und etwa daraus extrahierter Merkmale $\boldsymbol{c}$ in Abschnitt 3.8.2.

Zentrierung der transformierten Muster

Es bleibt nun noch die Klärung der Frage, wie die Bedingung (3.8.57) der Mittelwertfreiheit von $\boldsymbol{\phi}(\boldsymbol{f})$ zu berücksichtigen ist. Der Ansatz dafür beruht auf der in (3.8.65) eingeführten Matrix $\boldsymbol{T}$ der Skalarprodukte $\left(\boldsymbol{\phi}({}^i\boldsymbol{f})^\mathsf{T}\boldsymbol{\phi}({}^j\boldsymbol{f})\right)$. Wenn also die Elemente des Skalarproduktes zentriert (mittelwertfrei gemacht) werden können, ist das Problem gelöst.

Für die transformierten Muster wird i. Allg. gelten

$$E\{\boldsymbol{\phi}(\boldsymbol{f})\} \approx \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\phi}({}^j\boldsymbol{f}) = \boldsymbol{m}_{\boldsymbol{\phi}} \neq \boldsymbol{0}\ . \tag{3.8.70}$$

Dann ist offensichtlich

$$\boldsymbol{\phi}'(\boldsymbol{f}) = \boldsymbol{\phi}(\boldsymbol{f}) - \boldsymbol{m}_{\boldsymbol{\phi}} \tag{3.8.71}$$

mittelwertfrei. Die zugehörige Kovarianzmatrix ist in Analogie zu (3.8.60)

$$\begin{aligned}\boldsymbol{S}' &= \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\phi}'({}^j\boldsymbol{f})\,\boldsymbol{\phi}'({}^j\boldsymbol{f})^\mathsf{T} \\ &= \frac{1}{N}\sum_{j=1}^{N}\left(\boldsymbol{\phi}({}^j\boldsymbol{f}) - \boldsymbol{m}_{\boldsymbol{\phi}}\right)\left(\boldsymbol{\phi}({}^j\boldsymbol{f}) - \boldsymbol{m}_{\boldsymbol{\phi}}\right)^\mathsf{T}\ .\end{aligned} \tag{3.8.72}$$

Mit dem Vektor $\boldsymbol{\phi}'(\boldsymbol{f})$ und der Matrix S' kann man dann (3.8.60) – (3.8.66) analog durchgehen und kommt auf das Eigenwertproblem

$$N\vartheta'_\nu\boldsymbol{\beta}'_\nu = \boldsymbol{T}'\boldsymbol{\beta}'_\nu\ , \tag{3.8.73}$$

wobei die Matrix $\boldsymbol{T}'$ die Elemente $\left(\boldsymbol{\phi}'({}^i\boldsymbol{f})^\mathsf{T}\boldsymbol{\phi}'({}^j\boldsymbol{f})\right)$ hat. Die Lösungen $\boldsymbol{\beta}'_\nu$ werden analog zu (3.8.68) normiert auf $1 = (1/N)\vartheta'_\nu\left({\boldsymbol{\beta}'_\nu}^\mathsf{T}\boldsymbol{\beta}'_\nu\right)$. Für die Elemente von $\boldsymbol{T}'$ gilt offensichtlich (t_{ij} sind die Elemente der Matrix $\boldsymbol{T}$)

$$\begin{aligned}t'_{ij} &= \left(\boldsymbol{\phi}'({}^i\boldsymbol{f})^\mathsf{T}\boldsymbol{\phi}'({}^j\boldsymbol{f})\right) \\ &= \left(\left(\boldsymbol{\phi}({}^i\boldsymbol{f}) - \boldsymbol{m}_{\boldsymbol{\phi}}\right)^\mathsf{T}\left(\boldsymbol{\phi}({}^j\boldsymbol{f}) - \boldsymbol{m}_{\boldsymbol{\phi}}\right)\right) \\ &= t_{ij} - \boldsymbol{\phi}({}^i\boldsymbol{f})^\mathsf{T}\boldsymbol{m}_{\boldsymbol{\phi}} - \boldsymbol{m}_{\boldsymbol{\phi}}^\mathsf{T}\boldsymbol{\phi}({}^j\boldsymbol{f}) + \boldsymbol{m}_{\boldsymbol{\phi}}^\mathsf{T}\boldsymbol{m}_{\boldsymbol{\phi}} \qquad (3.8.74) \\ &= t_{ij} - \frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{\phi}({}^i\boldsymbol{f})^\mathsf{T}\boldsymbol{\phi}({}^k\boldsymbol{f})\right) - \frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{\phi}({}^k\boldsymbol{f})^\mathsf{T}\boldsymbol{\phi}({}^j\boldsymbol{f})\right) \\ &\quad + \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\left(\boldsymbol{\phi}({}^k\boldsymbol{f})^\mathsf{T}\boldsymbol{\phi}({}^l\boldsymbol{f})\right)\ . \qquad (3.8.75)\end{aligned}$$

Es wird nochmals betont, dass das Ziel bei der nichtlinearen Hauptachsentransformation darin besteht, die Berechnung der transformierten Vektoren $\boldsymbol{\phi}({}^k\boldsymbol{f})$ überhaupt zu vermeiden. Daher ist *nicht* (3.8.74) die gesuchte Lösung, sondern (3.8.75), da dort nur Skalarprodukte der transformierten Vektoren auftreten. Die beiden Summen über N in (3.8.75) brauchen für eine N^2 Matrix nur N-mal, die Doppelsumme nur einmal ausgewertet zu werden.

3.8.4 Optimale lineare Transformationen

Unter dieser Überschrift fassen wir lineare Transformationen zusammen, die nicht notwendig auch orthogonal sind und die ein auf dem Klassifikationsrisiko basierendes Gütekriterium optimieren. Die bisherige Vorgehensweise, die Merkmalsgewinnung unabhängig vom Klassifikator betrachtete, ist nun nicht mehr möglich. Dafür ergibt sich der Vorteil, dass man Merkmale erhält, die speziell auf einen Klassifikatortyp zugeschnitten sind; daher wird diese Vorgehensweise auch als **klassifikatorbezogene Merkmalsauswahl** bezeichnet. Da Klassifikatoren erst im nächsten Kapitel behandelt werden, ist hier ein Vorgriff auf dort abgeleitete Ergebnisse erforderlich. Danach ist der sogenannte BAYES-Klassifikator ein sehr allgemeines Konzept, das die Minimierung der mittleren Kosten oder des Risikos bei der Klassifikation erlaubt. Je nach Anwendungsfall kann man die Kosten unterschiedlicher Fehlklassifikationen und Rückweisungen geeignet wählen. Nach (4.1.10), S. 309, ist das minimale Risiko durch

$$V(\delta^*) = \sum_{\kappa=1}^{k} p_\kappa \sum_{\lambda=0}^{k} r_{\lambda\kappa} \int_{\mathbb{R}_{\boldsymbol{c}}} p(\boldsymbol{c}\,|\,\Omega_\kappa)\delta^*(\Omega_\lambda\,|\,\boldsymbol{c}\,)\,\mathrm{d}\boldsymbol{c} \tag{3.8.76}$$

definiert, wobei δ^* die optimale Entscheidungsregel gemäß Definition 4.2, S. 309, ist und die sonstigen in dieser Gleichung auftretenden Größen in Abschnitt 4.1 erläutert werden. Durch geeignete Wahl der Terme $r_{\lambda\kappa}$ geht das Risiko V in die Fehlerwahrscheinlichkeit p_f des Klassifikators über (s. Abschnitt 4.1.4 und (4.1.31), S. 314). Sind die Merkmale $\boldsymbol{c}$ mit (3.2.2) berechnet worden, so ist das Risiko eine Funktion von $\boldsymbol{\Phi}$, und das Problem besteht darin, die Transformationsmatrix $\boldsymbol{\Phi}$ zu berechnen, die das Risiko V minimiert. Man hätte dann eine bezüglich des Risikos *optimale lineare* Transformation zur Merkmalsgewinnung. Im Prinzip ist auch eine nichtlineare Transformation von der Form

$$\boldsymbol{c} = \boldsymbol{\varphi}(\boldsymbol{f}, \boldsymbol{a}) \tag{3.8.77}$$

möglich, in der $\boldsymbol{\varphi}$ eine parametrische Familie von Funktionen und $\boldsymbol{a}$ ein Parametervektor ist. In diesem Falle ist $\boldsymbol{a}$ so zu bestimmen, dass V minimiert wird. Bedingungen für die Existenz optimaler Parameter $\boldsymbol{a}$ sind bekannt; sie sind so allgemein, dass sie als nicht kritisch zu betrachten sind. Das Problem liegt jedoch in der tatsächlichen Berechnung des Parametervektors $\boldsymbol{a}$ bzw. der Transformationsmatrix $\boldsymbol{\Phi}$. Selbst zur Berechnung der linearen Transformation sind weitere einschränkende Annahmen zu treffen. Eines der bei der Lösung auftretenden Probleme besteht darin, dass man die bedingte Dichte $p(\boldsymbol{f}\,|\,\Omega_\kappa)$ der Muster kennen oder ermitteln muss und dass man die zugehörige Dichte $p(\boldsymbol{c}\,|\,\Omega_\kappa)$ berechnen muss, wenn $\boldsymbol{c} = \boldsymbol{\Phi}\boldsymbol{f}$ ist. Diese Diskussion sollte die mit einem allgemeinen theoretischen Ansatz verbundenen Probleme aufzeigen.

Wenn die Komponenten des Merkmalsvektors $\boldsymbol{c}$ klassenweise *normalverteilt* sind, erhält man als *Prüfgrößen* u'_κ des Klassifikators, der die Fehlerwahrscheinlichkeit minimiert (s. (4.2.118), S. 349)

$$u'_\kappa(\boldsymbol{c}\,) = (\boldsymbol{c} - \boldsymbol{\mu}_\kappa)^\mathsf{T}\boldsymbol{\Sigma}_\kappa^{-1}(\boldsymbol{c} - \boldsymbol{\mu}_\kappa) + \gamma_\kappa\ . \tag{3.8.78}$$

Dabei sind $\boldsymbol{\mu}_\kappa$ und $\boldsymbol{\Sigma}_\kappa$ die durch Klasse Ω_κ, $\kappa = 1, \ldots, k$ bedingten Mittelwerte und Kovarianzmatrizen der Merkmale. Der Klassifikator entscheidet sich für die Klasse mit minimaler Prüfgröße u'_κ. Eine Vereinfachung ergibt sich, wenn man die klassenspezifische Konstante vernachlässigt. Die Prüfgrößen dieses in Abschnitt 4.2.5 als modifizierter **Minimumabstandsklassifikator** (MMA) bezeichneten Klassifikators sind

$$u_\kappa(\boldsymbol{c}) = (\boldsymbol{c} - \boldsymbol{\mu}_\kappa)^\mathsf{T} \boldsymbol{\Sigma}_\kappa^{-1} (\boldsymbol{c} - \boldsymbol{\mu}_\kappa) \; . \tag{3.8.79}$$

Sie lassen sich als Abstandsquadrate zwischen dem Merkmal $\boldsymbol{c}$ und bedingten Mittelwerten (oder Klassenzentren) $\boldsymbol{\mu}_\kappa$ auffassen. Natürlich lässt sich der MMA auch anwenden, wenn die Merkmale nicht normalverteilt sind, jedoch wird seine Leistung dann i. Allg. entsprechend geringer sein. Unabhängig von der Verteilungsdichte der $\boldsymbol{c}$ gilt als Verallgemeinerung der TSCHEBYSCHEFF-**Ungleichung**

$$P(u_\kappa \geq \alpha) < \frac{n}{\alpha} \; , \tag{3.8.80}$$

wobei n die Zahl der Komponenten von $\boldsymbol{c}$ ist. Diese Abschätzung der Wahrscheinlichkeit, dass u_κ eine vorgegebene Schranke überschreitet, lässt sich zur Abschätzung der Fehlerwahrscheinlichkeit des MMA heranziehen, und damit hat man ein Kriterium zur Merkmalsgewinnung. Im Merkmalsraum definiert u_κ ein Hyperellipsoid mit dem „Radius“ α. Die durch den MMA bestimmte Grenze $H_{\kappa\lambda}$ zwischen zwei Klassen Ω_κ und Ω_λ ergibt sich aus der Gleichung

$$H_{\kappa\lambda} : \; (\boldsymbol{c} - \boldsymbol{\mu}_\kappa)^\mathsf{T} \boldsymbol{\Sigma}_\kappa^{-1} (\boldsymbol{c} - \boldsymbol{\mu}_\kappa) = (\boldsymbol{c} - \boldsymbol{\mu}_\lambda)^\mathsf{T} \boldsymbol{\Sigma}_\lambda^{-1} (\boldsymbol{c} - \boldsymbol{\mu}_\lambda) \; . \tag{3.8.81}$$

Ein Beispiel für solche Klassengrenzen zeigt Bild 3.8.7. Man entnimmt dem Bild, dass ein Muster aus Ω_2 mit Sicherheit *richtig* klassifiziert wird, wenn es innerhalb der Ellipse, welche die Klassengrenze H_{12} zwischen Ω_1 und Ω_2 berührt, liegt. Auf dieser Ellipse ist $u_2 = \text{const}$. Definiert man allgemein eine Konstante $u_{\kappa\lambda}$ als kleinsten Abstand aller Punkte auf der Klassengrenze $H_{\kappa\lambda}$ vom Mittelwert μ_κ, so wird ein Muster aus Ω_κ mit Sicherheit richtig klassifiziert, wenn es innerhalb des Ellipsoids mit dem Radius

$$u_{\kappa m} = \min_{\lambda \neq \kappa} u_{\kappa\lambda} \tag{3.8.82}$$

liegt. Die Zahlen $u_{\kappa\lambda}$ erhält man aus

$$u_{\kappa\lambda} = \min_{\{\boldsymbol{c} \in H_{\kappa\lambda}\}} u_\kappa \; , \quad \lambda = 1, \ldots, k \quad \lambda \neq \kappa \; . \tag{3.8.83}$$

Die Wahrscheinlichkeit $p_{f\kappa}$, dass man Muster aus der Klasse Ω_κ *falsch* klassifiziert, ist also sicher nicht größer als die Wahrscheinlichkeit, dass Muster *außerhalb* der Ellipse mit Radius $u_{\kappa m}$ liegen. Damit erhält man mit (3.8.80) als Abschätzung der bedingten Fehlerwahrscheinlichkeit des MMA

$$p_{f\kappa} \leq P(u_\kappa > u_{\kappa m}) < \frac{n}{u_{\kappa m}} \; , \quad p_{f\kappa} < \frac{n}{u_{\kappa m}} \; . \tag{3.8.84}$$

Ein geeignetes Kriterium zur Beurteilung der Güte von Merkmalen, die mit einem MMA klassifiziert werden, ist demnach

$$s_6 = \sum_{\kappa=1}^{k} p_\kappa \frac{n}{u_{\kappa m}} \; . \tag{3.8.85}$$

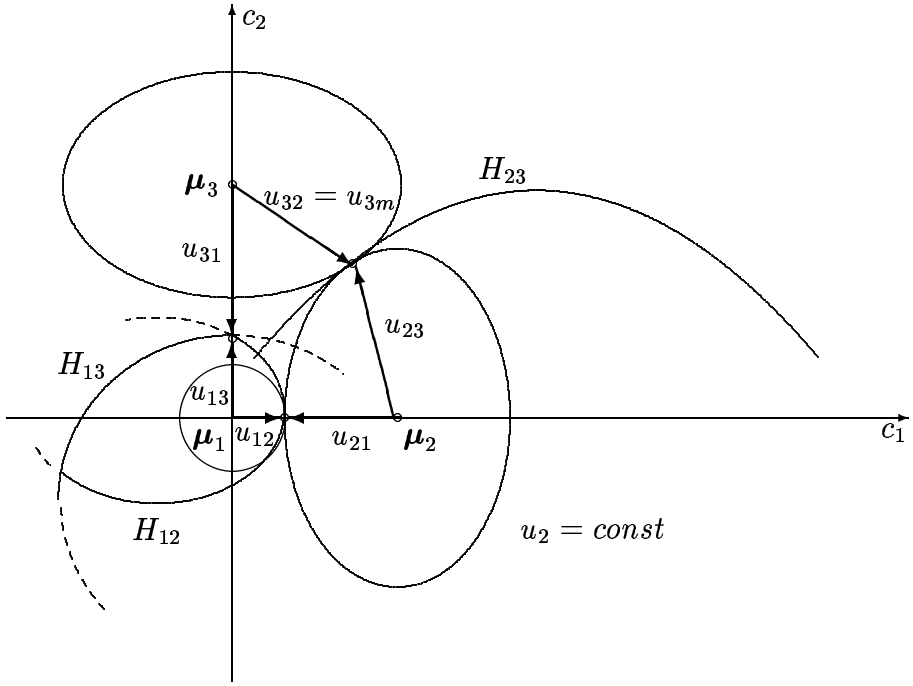


Bild 3.8.7: Klassengrenzen $H_{\kappa\lambda}$ eines Minimumabstandsklassifikators und Bedeutung der $u_{\kappa m}$ und $u_{\kappa\lambda}$ in (3.8.82) und (3.8.83). In diesem speziellen Fall ist $u_{21} = u_{23} = u_{2m}$, aber z. B. $u_{12} < u_{13}$, also $u_{12} = u_{1m}$. Muster aus Ω_1, die innerhalb des Kreises mit Radius u_{1m} liegen, werden mit Sicherheit richtig klassifiziert

Gesucht ist die Transformationsmatrix $\boldsymbol{\Phi}$, welche s_6 minimiert, und im Folgenden wird gezeigt, wie $\boldsymbol{\Phi}$ zu berechnen ist.

Zunächst überzeugt man sich leicht, dass die Größen $u_\kappa(\boldsymbol{c})$ in (3.8.79) *invariant* gegenüber einer linearen Transformation der Merkmale $\boldsymbol{c}$ mit einer regulären Matrix sind. Ist $\boldsymbol{\Phi}$ eine *optimale* Matrix, die s_6 minimiert, so erhält man Merkmale aus

$$\boldsymbol{c} = \boldsymbol{\Phi}\boldsymbol{f} \ .$$

Ist $\boldsymbol{B}$ eine reguläre n^2 Matrix, so ergibt sich für Merkmale

$$\boldsymbol{c}' = \boldsymbol{B}\boldsymbol{c} = \boldsymbol{B}\boldsymbol{\Phi}\boldsymbol{f} = \widetilde{\boldsymbol{\Phi}}\boldsymbol{f} \tag{3.8.86}$$

der *gleiche* Wert von u_κ und damit von s_6 wie für Merkmale $\boldsymbol{c}$; d. h. auch $\widetilde{\boldsymbol{\Phi}}$ ist eine optimale Matrix. Wenn die n Zeilen von $\boldsymbol{\Phi}$ linear unabhängig sind, definiert man eine Matrix $\boldsymbol{B}^{-1}$, welche z. B. die ersten n Spalten von $\boldsymbol{\Phi}$ enthält. Dann gilt

$$\begin{aligned}
\widetilde{\boldsymbol{\Phi}} &= \boldsymbol{B}\boldsymbol{\Phi} \\
&= \boldsymbol{B}(\boldsymbol{B}^{-1}|\boldsymbol{\Phi}_{\mathrm{Rest}}) \\
&= (\boldsymbol{I}|\boldsymbol{B}\boldsymbol{\Phi}_{\mathrm{Rest}}) \\
&= \begin{pmatrix}
1 & 0 & \dots & 0 & \varphi_{1,n+1} & \dots & \varphi_{1,M} \\
0 & 1 & \dots & 0 & \varphi_{2,n+1} & \dots & \varphi_{2,M} \\
\vdots & & & & & & \\
0 & 0 & \dots & 1 & \varphi_{n,n+1} & \dots & \varphi_{n,M}
\end{pmatrix} .
\end{aligned} \tag{3.8.87}$$

Zu einer Lösung $\mathbf{\Phi}$, die s_6 in (3.8.85) minimiert, lässt sich also eine äquivalente Matrix $\widetilde{\mathbf{\Phi}}$ angeben, die ebenfalls s_6 minimiert, aber statt nM unbekannter Elemente nur $n(M-n)$ unbekannte Elemente enthält. Statt der Matrix $\mathbf{\Phi}$ wird man also direkt $\widetilde{\mathbf{\Phi}}$ bestimmen.

Die Berechnung der Matrix $\widetilde{\mathbf{\Phi}}$ kann im Prinzip mit den bekannten Optimierungsverfahren durchgeführt werden. Als Beispiel wird hier der **Koordinatenabstieg** verwendet. Dabei wird in einem Optimierungsschritt nur das Minimum von s_6 bezüglich *eines* Elementes φ_{ij} von $\widetilde{\mathbf{\Phi}}$ (oder bezüglich *einer* Koordinate) bestimmt. Alle Elemente $\varphi_{ij},\ i = 1,\ldots,n$ und $j = n+1,\ldots,M$ werden in fester Reihenfolge so oft durchlaufen, bis das Kriterium s_6 sich nicht mehr verringert, also ein relatives Minimum gefunden wurde. Die Bestimmung des Minimums von s_6 bezüglich einer Koordinate erfolgt näherungsweise, indem man das betrachtete Element φ_{ij} versuchsweise vergrößert und verkleinert und die dadurch verursachte Änderung von s_6 untersucht.

Der Algorithmus zur Bestimmung von $\widetilde{\mathbf{\Phi}}$ in (3.8.87) zur linearen Merkmalsgewinnung gemäß $\boldsymbol{c} = \widetilde{\mathbf{\Phi}}\boldsymbol{f}$ arbeitet wie folgt:

Es ist $\dim(\boldsymbol{c}) = n$ und $\dim(\boldsymbol{f}) = M$. Weiterhin sei $\boldsymbol{m}_\kappa = E_\kappa\{\boldsymbol{f}\}$ und $\boldsymbol{L}_\kappa = E_\kappa\{(\boldsymbol{f}-\boldsymbol{m})(\boldsymbol{f}-\boldsymbol{m})^\mathsf{T}\}$.

1. Anfangswert $\widetilde{\mathbf{\Phi}}^{(0)}$ von $\widetilde{\mathbf{\Phi}}$ ist

$$\widetilde{\mathbf{\Phi}}^{(0)} = (\boldsymbol{I}_n|\mathbf{0})\ . \tag{3.8.88}$$

2. Führe die Schritte 3 – 11 für $j = n+1, n+2,\ldots,M$ und $i = 1,2,\ldots,n$ aus, d. h. spaltenweise von oben nach unten und dann links nach rechts in (3.8.87):
3. Der Iterationschritt ist $l = i + (j-n-1)n - 1$.
4. $\boldsymbol{c} = \widetilde{\mathbf{\Phi}}^{(l)}\boldsymbol{f}\ ; \quad \boldsymbol{\mu}_\kappa = \widetilde{\mathbf{\Phi}}^{(l)}\boldsymbol{m}_\kappa\ ; \quad \mathbf{\Sigma}_\kappa = \mathbf{\Phi}^{(l)}\boldsymbol{L}_\kappa{\mathbf{\Phi}^{(l)}}^\mathsf{T}$.
5. Berechne $u_{\kappa\lambda}$ gemäß (3.8.83), siehe dazu den unten angegebenen Algorithmus, $\kappa,\lambda = 1,\ldots,k$.
6. Berechne $u_{\kappa m}$ gemäß (3.8.82).
7. Berechne $s_6 = s_6(\widetilde{\mathbf{\Phi}}^{(l)})$ gemäß (3.8.85).
8. Ersetze φ_{ij} in $\widetilde{\mathbf{\Phi}}^{(l)}$ durch $\varphi_{ij} + mh$ mit $m = \pm 1$ und $h = \text{const} \approx 0,1$ und bezeichne die so entstehende Matrix mit $\widetilde{\mathbf{\Phi}}_m^{(l)}$.
9. Ist $s_6(\widetilde{\mathbf{\Phi}}_0^{(l)}) \le s_6(\widetilde{\mathbf{\Phi}}_{\pm 1}^{(l)})$, so nimm im nächsten Iterationschritt $l+1$ die Matrix $\widetilde{\mathbf{\Phi}}^{(l)}$.
10. Ist $s_6(\widetilde{\mathbf{\Phi}}_1^{(l)}) < s_6(\widetilde{\mathbf{\Phi}}_{-1}^{(l)})$, so berechne $s_6(\widetilde{\mathbf{\Phi}}_m^{(l)})$ für $m = 1,2,\ldots,L$ $(L \approx 10)$ und nimm im nächsten Iterationschritt $l+1$ die Matrix, die s_6 minimiert.
11. Ist $s_6(\widetilde{\mathbf{\Phi}}_{-1}^{(l)}) < s_6(\widetilde{\mathbf{\Phi}}_1^{(l)})$, so berechne $s_6(\widetilde{\mathbf{\Phi}}_m^{(l)})$ für $m = -1,-2,\ldots,-L$ und nimm im nächsten Iterationschritt $l+1$ die Matrix, die s_6 minimiert.
12. Wiederhole ab Schritt 2 bis sich entweder nichts mehr ändert oder eine vorgegebene Zahl von Wiederholungen erreicht ist.

Obiger Algorithmus durchläuft die Matrixelemente nur wenige Male, um die Rechenzeit zu begrenzen.

Ein Problem ist noch die Berechnung der $u_{\kappa\lambda}$ in Schritt 5 des Algorithmus. Diese ist mit der Methode der projizierten Gradienten möglich. Das Verfahren wird zunächst informell an Bild 3.8.8 erläutert. Es wird ein Startpunkt $\boldsymbol{c}_0$ auf der Klassengrenze $H_{\kappa\lambda}$ bestimmt, den man z. B. als Schnittpunkt der Verbindungsgeraden zwischen $\boldsymbol{\mu}_\kappa$ und $\boldsymbol{\mu}_\lambda$ mit $H_{\kappa\lambda}$ wählen kann. Dann wird ein Punkt $\boldsymbol{c}_0'$ ermittelt, der in der Hyperebene liegt, welche $H_{\kappa\lambda}$ in $\boldsymbol{c}_0$ berührt, und zwar

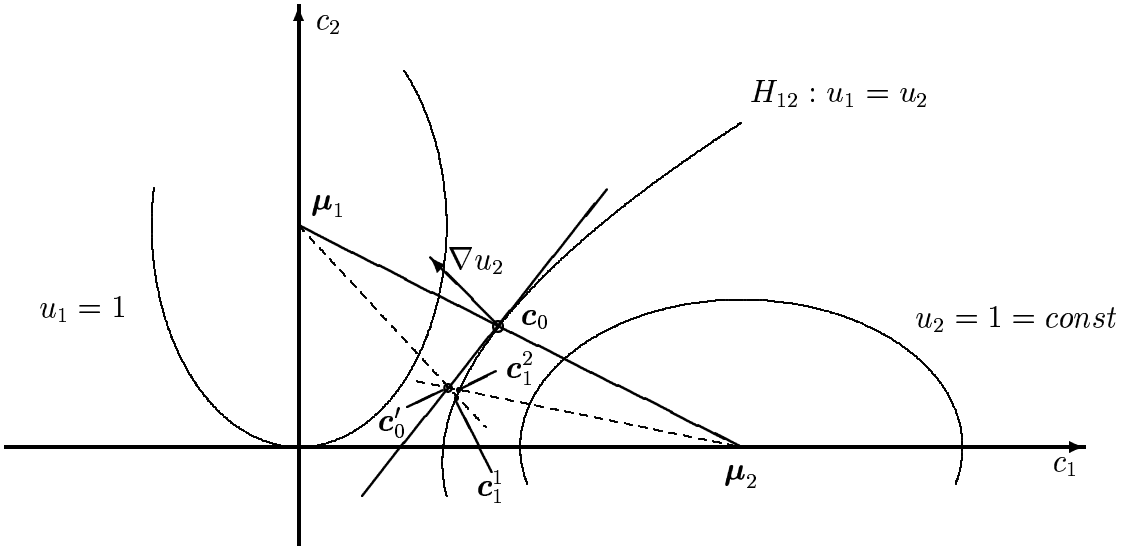


Bild 3.8.8: Zur Methode der projizierten Gradienten. Die Trennlinie H_{12} wurde hier nicht maßstäblich gezeichnet, da nur das Prinzip verdeutlicht werden soll

wird $\boldsymbol{c'}_0$ nur auf der Geraden gesucht, die durch Projektion des Gradienten von $u_\kappa(\boldsymbol{c})$ auf diese Hyperebene definiert ist. Der so eingeschränkte Punkt $\boldsymbol{c}_0$ muss $u_\kappa(\boldsymbol{c})$ in (3.8.79) minimieren, d. h. , er liegt in Richtung der negativen Projektion des Gradienten. Schließlich wird ein Punkt $\boldsymbol{c}_1$ als Schnittpunkt der Verbindungsgeraden zwischen $\boldsymbol{c'}_0$ und $\boldsymbol{\mu}_\kappa$ mit $H_{\kappa\lambda}$ bestimmt. Dieser Punkt $\boldsymbol{c}_1$ liegt also auf der Klassengrenze $H_{\kappa\lambda}$ und es ist $u_\kappa(\boldsymbol{c}_1) \leq u_\kappa(\boldsymbol{c}_0)$. Da es Fälle geben kann, in denen ein solcher Schnittpunkt nicht existiert, wird die Rechnung sowohl für $\boldsymbol{\mu}_\kappa$, $u_\kappa(\boldsymbol{c})$ als auch $\boldsymbol{\mu}_\lambda$, $u_\lambda(\boldsymbol{c})$ ausgeführt, da wenigstens bei einem Rechengang eine Lösung $\boldsymbol{c}_1$ existiert. Gibt es in jedem Rechengang einen Schnittpunkt, so wird der gewählt, der den kleinsten Wert für u_κ liefert. Im Bild sind diese beiden Punkte mit $\boldsymbol{c}_1^1$ und $\boldsymbol{c}_1^2$ bezeichnet. Das Verfahren ist deshalb möglich, weil die Minimierung von $u_\kappa(\boldsymbol{c})$ und $u_\lambda(\boldsymbol{c})$ mit der gemeinsamen Nebenbedingung (3.8.81) die gleiche Lösung ergibt.

Algorithmus zur Bestimmung der $u_{\kappa\lambda}$ gemäß (3.8.83) nach der Methode der projizierten Gradienten:

5.1 Zur Bestimmung des Startpunktes $\boldsymbol{c}_0$ wird zunächst die Verbindungsgerade zwischen $\boldsymbol{\mu}_\kappa$ und $\boldsymbol{\mu}_\lambda$ angegeben. Sie hat die Gleichung

$$\boldsymbol{c}(\theta) = \boldsymbol{\mu}_\kappa + \theta(\boldsymbol{\mu}_\lambda - \boldsymbol{\mu}_\kappa) \; . \tag{3.8.89}$$

Um den Schnittpunkt dieser Geraden mit der Klassengrenze $H_{\kappa\lambda}$ zu berechnen, setzt man (3.8.89) in (3.8.81) ein und löst die quadratische Gleichung

$$\begin{aligned} 0 &= \theta^2 \boldsymbol{a}^\mathsf{T}(\boldsymbol{\Sigma}_\kappa^{-1} - \boldsymbol{\Sigma}_\lambda^{-1})\boldsymbol{a} + 2\theta \boldsymbol{a}^\mathsf{T}\boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{a} - \boldsymbol{a}^\mathsf{T}\boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{a} \; , \\ \boldsymbol{a} &= \boldsymbol{\mu}_\lambda - \boldsymbol{\mu}_\kappa \end{aligned} \tag{3.8.90}$$

nach θ auf. Die reelle Lösung θ_1 mit $0 < \theta_1 < 1$ ergibt mit (3.8.89) den Startpunkt

$$\boldsymbol{c}_0 = \boldsymbol{c}(\theta_1) = \boldsymbol{\mu}_\kappa + \theta_1(\boldsymbol{\mu}_\lambda - \boldsymbol{\mu}_\kappa) \; . \tag{3.8.91}$$

5.2 Der Gradient der zu minimierenden Funktion u_κ im Punkt $\boldsymbol{c}_0$ ist

$$\nabla\, u_\kappa(\boldsymbol{c}_0) = 2\boldsymbol{\Sigma}_\kappa^{-1}(\boldsymbol{c}_0 - \boldsymbol{\mu}_\kappa)\ . \tag{3.8.92}$$

5.3 Die Projektion des Gradienten auf die Hyperebene, welche $H_{\kappa\lambda}$ in $\boldsymbol{c}_0$ berührt, erhält man aus

$$\boldsymbol{r} = \boldsymbol{P} \cdot \nabla\, u_\kappa(\boldsymbol{c}_0)\ . \tag{3.8.93}$$

Dabei ist $\boldsymbol{P}$ die Projektionsmatrix

$$\boldsymbol{P} = \boldsymbol{I} - \frac{\boldsymbol{n}\boldsymbol{n}^\mathsf{T}}{\boldsymbol{n}^\mathsf{T}\boldsymbol{n}}\ , \tag{3.8.94}$$

und $\boldsymbol{n}$ ist der Normalenvektor der Hyperebene

$$\boldsymbol{n} = 2\boldsymbol{\Sigma}_\kappa^{-1}(\boldsymbol{c}_0 - \boldsymbol{\mu}_\kappa) - 2\boldsymbol{\Sigma}_\lambda^{-1}(\boldsymbol{c}_0 - \boldsymbol{\mu}_\lambda)\ . \tag{3.8.95}$$

5.4 Man bestimme nun das Minimum von $u_\kappa(\boldsymbol{c}_0 + \theta\boldsymbol{r}\,)$, indem man die Gleichung

$$\frac{du_\kappa(\boldsymbol{c}_0 + \theta\boldsymbol{r})}{d\theta} = 0 \tag{3.8.96}$$

nach θ auflöst. Man erhält die Gleichung

$$\theta_0 = \frac{\boldsymbol{r}^\mathsf{T}\boldsymbol{\Sigma}_\kappa^{-1}(\boldsymbol{c}_0 - \boldsymbol{\mu}_\kappa)}{\boldsymbol{r}^\mathsf{T}\boldsymbol{\Sigma}_\kappa^{-1}\boldsymbol{r}}\ . \tag{3.8.97}$$

5.5 Mit θ_0 ergibt sich der gesuchte Lösungspunkt $\boldsymbol{c'}$ zu

$$\boldsymbol{c'}_0 = \boldsymbol{c}_0 + \theta_0\boldsymbol{r}\ . \tag{3.8.98}$$

5.6 Nun wird der neue Punkt $\boldsymbol{c}_1$ auf der Klassengrenze $H_{\kappa\lambda}$ bestimmt. Man erhält ihn entsprechend dem Verfahren in Schritt 5.1, wenn man $\boldsymbol{\mu}_\lambda$ durch $\boldsymbol{c}'_0$ ersetzt.

5.7 Die Rechnungen ab Schritt 5.1 werden auch für $u_\lambda(\boldsymbol{c}\,)$ durchgeführt.

5.8 Von den in beiden Rechnungsgängen gewonnenen Punkten $\boldsymbol{c}_1$ wird der ausgewählt, der $u_\kappa(\boldsymbol{c}_1)$ minimiert.

5.9 Die Schritte 5.1–8 werden wiederholt, bis sich $u_\kappa(\boldsymbol{c}\,)$ kaum noch verändert. Der zugehörige Punkt auf der Klassengrenze $H_{\kappa\lambda}$ sei $\boldsymbol{c}^*$. Dann gilt in (3.8.83)

$$u_{\kappa\lambda} = u_\kappa(\boldsymbol{c}^*) \qquad \text{und} \qquad u_{\lambda\kappa} = u_\lambda(\boldsymbol{c}^*)\ . \tag{3.8.99}$$

Es wurde bereits erwähnt, dass die Matrix $\widetilde{\boldsymbol{\Phi}}$ in (3.8.87) $n(M-n)$ unbekannte Elemente enthält. Um diese Zahl zu reduzieren, kann man statt des Musters $\boldsymbol{f}$ mit M Koeffizienten auch eine Approximation von $\boldsymbol{f}$ mit $M' < M$ Koeffizienten verwenden. Dafür bietet sich z. B. die KARHUNEN–LOÈVE oder Hauptachsentransformation an, die im vorigen Abschnitt definiert wurde. Das hat den weiteren Vorteil, dass diese Koeffizienten näherungsweise normalverteilt und daher für den MMA besser geeignet sind als das ursprüngliche Muster $\boldsymbol{f}$. Zwar wurde hier explizit nur der MMA diskutiert. Es ist aber offensichtlich, dass die Schritte 8. bis 11. des Algorithmus auf jeden Klassifikator anwendbar sind. Die Abschätzung s_6 ist durch eine geeignete andere zu ersetzen, beispielsweise eine direkte Schätzung der Fehlerwahrscheinlichkeit wie in (3.9.9), (3.9.10). Sicherlich ist der Rechenaufwand erheblich, aber das, was praktisch berechenbar ist, hängt vor allem vom Stand der Rechnertechnologie ab, die ständig verbessert wird.

3.8.5 Bemerkungen

Die Idee, Merkmale zu bestimmen, welche die Fehlerwahrscheinlichkeit minimieren, ist zunächst äußerst attraktiv. Der Algorithmus zur Bestimmung solcher Merkmale für den MMA gibt einen Eindruck von dem dafür erforderlichen Rechenaufwand und den daraus resultierenden praktischen Grenzen. Der Rechenaufwand für lineare Transformationen zur Merkmalsgewinnung ist für die FOURIER- und WALSH-Transformation in Abschnitt 3.2.2, 3.2.5 am geringsten, da es schnelle Algorithmen dafür gibt und die Transformationsmatrix für alle Problemkreise dieselbe ist (bei festem M). Für die Hauptachsentransformation und ähnliche problemabhängige Verfahren in Abschnitt 3.8.2 gibt es keine schnellen Algorithmen und die Transformationsmatrix $\mathbf{\Phi}$ muss für jeden Problemkreis neu berechnet werden. Der Rechenaufwand vergrößert sich, bleibt aber unproblematisch, solange man einfache Muster klassifizieren will, bei denen die Zahl der Abtastwerte bei etwa $M = 300$ bis 3000 liegt. Bei den optimalen Transformationen von Abschnitt 3.8.4 wird die Berechnung der Transformationsmatrix zu einem echten Problem aufgrund der Rechenzeit, obwohl man diese Berechnung für jedes System nur einmal vorweg durchzuführen hat.

Ein experimenteller Vergleich von verschiedenen Transformationen kann durchgeführt werden, indem man für eine Stichprobe ω von Mustern verschiedene Verfahren der Merkmalsgewinnung realisiert und mit Hilfe eines Klassifikators die Fehlerrate $\widehat{p}_{\mathrm{f}}$, also einen Schätzwert der Fehlerwahrscheinlichkeit, berechnet. Dieses ist in Bild 3.8.9 für zwei verschiedene Stichproben dargestellt.

Bild 3.8.9a zeigt das Ergebnis für eine Stichprobe mit etwa 22.000 handgedruckten Ziffern der Klassen „0“ bis „9“. Die Ziffern wurden in einem 16×12 Raster größennormiert dargestellt und zufällig in eine Lernstichprobe von etwa 10.000 und eine Teststichprobe von etwa 12.000 Mustern zerlegt. In Bild 3.8.9b ist das Ergebnis für eine Stichprobe mit etwa 15.000 isoliert gesprochenen Ziffern der Klassen „null“ bis „neun“ dargestellt. Die gesprochenen Ziffern waren als 14×20 Matrix in vorverarbeiteter Form gegeben, wobei die 14 Zeilen der Matrix Energieanteile je eines Terz-Bandfilters enthalten und die 20 Spalten eine Unterteilung der Dauer des Wortes in 20 gleichlange Zeitabschnitte ergeben. Auch hier wurde eine Lernstichprobe mit etwa 10.000 und eine Teststichprobe mit etwa 5.000 Mustern gebildet. Das Klassifikationssystem wurde mit der Lernstichprobe dimensioniert, Fehlerraten mit der Teststichprobe ermittelt. Als Klassifikator wurde der in Abschnitt 4.2.5 beschriebene optimale Klassifikator für normalverteilte Merkmale verwendet. Mit $\mathrm{DFT}_{2\mathrm{D}}$ bzw. $\mathrm{WHT}_{2\mathrm{D}}$ werden Merkmale gemäß (3.2.23) bzw. (3.2.59) bezeichnet, die man aus der zweidimensionalen FOURIER- bzw. WALSH-Transformation erhält. Aus den Koeffizientenmatrizen wurden die ersten n wie in Bild 3.2.3, S. 174, ausgewählt. Eine Merkmalsbewertung und –auswahl mit den Methoden von Abschnitt 3.9 wurde absichtlich nicht vorgenommen. Mit KLT werden in Bild 3.8.9 Merkmale bezeichnet, die s_1 in (3.8.1) maximieren, d. h. es ist die KARHUNEN–LOÈVE-Transformation, und mit DIV bzw. MMA werden Merkmale bezeichnet, die die mittlere Divergenz (3.9.15) maximieren bzw. s_6 in (3.8.85) minimieren. Bei MMA und DIV Merkmalen wurde, wie in Abschnitt 3.8.4 erwähnt, eine KL Transformation vorgeschaltet, d. h. statt der Abtastwerte $\boldsymbol{f}$ wurden in (3.2.2) die ersten 70 Koeffizienten der KL Transformation genommen.

Sicherlich kann ein experimenteller Vergleich keine allgemeine Aussage über die Güte von Verfahren zur Merkmalsgewinnung liefern, aber die Ergebnisse an den beiden umfangreichen Stichproben geben einen Eindruck von den Grenzen und relativen Vorteilen der Verfahren.

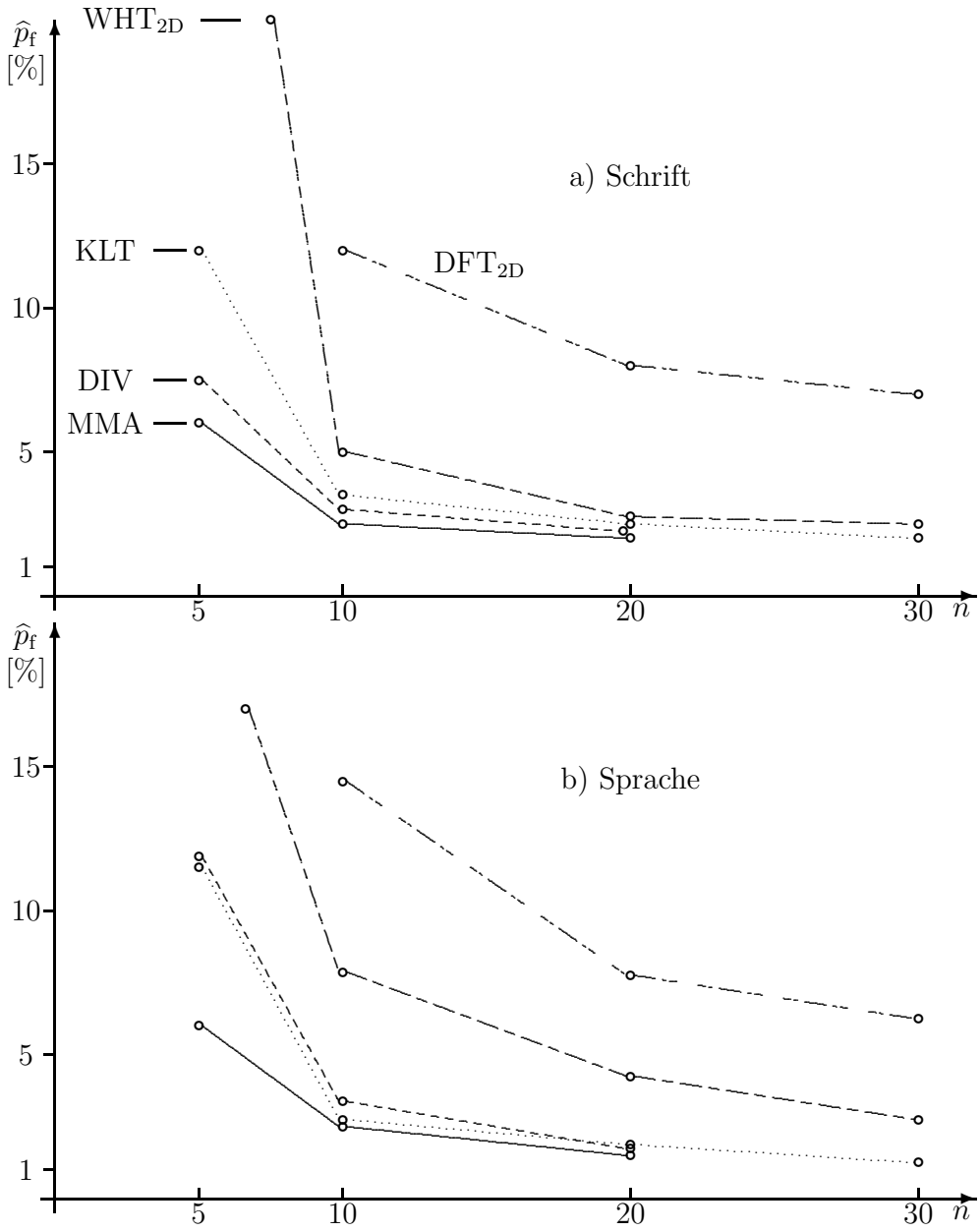


Bild 3.8.9: a) Klassifikation handgedruckter Ziffern mit einigen der im Text erläuterten Verfahren zur Merkmalsgewinnung, wobei n die Zahl der Merkmale und $\widehat{p}_\mathrm{f}$ die Fehlerrate ist. b) Klassifikation isoliert gesprochener Ziffern mit verschiedenen Merkmalen. (Für die Überlassung der dabei verwendeten Daten wird Herrn Prof. SCHÜRMANN, Forschungsinstitut der AEG-Telefunken in Ulm, sehr herzlich gedankt)

3.9 Merkmalsbewertung und –auswahl (VA.1.2.3, 13.04.2004)

3.9.1 Anliegen und Probleme

Mit den heuristischen Verfahren von Abschnitt 3.2 – Abschnitt 3.5 ist es relativ leicht möglich, eine große Zahl n' von Merkmalen zu erzeugen. Der Aufwand für die Klassifikation steigt mit der Zahl dieser Merkmale an. Das ist intuitiv unmittelbar klar und geht auch aus den speziellen Klassifikationsverfahren von Kapitel 4 hervor. Außerdem verursacht auch die Gewinnung jedes einzelnen Merkmals einen gewissen Aufwand. Aus diesen Gründen wird man stets bestrebt sein, dass die Zahl $n < n'$ der tatsächlich verwendeten Merkmale so klein wie möglich ist, um den Gesamtaufwand für die Klassifikation in erträglichen Grenzen zu halten. Damit ergibt sich die Aufgabe, eine Menge mit n' vorgegebenen Merkmalen durch eine **Merkmalsauswahl** auf eine Untermenge mit n „möglichst geeigneten“ Merkmalen zu reduzieren.

Definition 3.16 *Eine „beste“ Untermenge von Merkmalen hat die Eigenschaft, dass es keine andere mit höchstens genau so vielen Merkmalen gibt, wobei die Merkmale dieser anderen Untermenge eine Klassifikation mit* geringerer *Fehlerwahrscheinlichkeit erlauben.*

Aus zwei Gründen, die in den folgenden beiden Absätzen erläutert werden, ist es i. Allg. nicht möglich, diese beste Untermenge zu bestimmen. Daher muss man sich mit suboptimalen Ansätzen begnügen oder mit „möglichst geeigneten“ Merkmalen. Ein einwandfreies Kriterium zur Messung der Güte von Merkmalen ist die in einem bestimmten Klassifikationssystem erreichte Fehlerwahrscheinlichkeit, wie auch in Abschnitt 3.8.1 ausgeführt wurde. Um den Aufwand bei der Merkmalsauswahl zu reduzieren, werden jedoch meistens Kriterien oder Gütemaße verwendet, die *unabhängig* vom Klassifikator berechnet werden können. Beispiele für solche Gütemaße folgen im nächsten Abschnitt. Damit wird die *Bewertung* der Merkmale als eigenes Problem, ohne Beachtung der sonstigen Moduln des Klassifikationssystems, durchgeführt. Das vereinfacht das Problem, führt aber i. Allg. dazu, dass die so bestimmten Merkmale nicht die für das Gesamtsystem besten sind.

Auch wenn man annimmt, dass geeignete Maße zur Beurteilung der Güte von Merkmalen bekannt sind, ist die Bestimmung einer geeigneten Untermenge ein schwieriges Problem. Wegen der in der Regel vorhandenen statistischen Abhängigkeiten zwischen den Merkmalen müsste man bei einer vollständigen Suchmethode alle Untermengen beurteilen, um die optimale zu finden. Zu einer vorgegebenen Menge mit n' Merkmalen gibt es genau $\binom{n'}{n}$ verschiedene Untermengen mit $n < n'$ Merkmalen. Hat man beispielsweise $n' = 300$ Merkmale vorgegeben und will aus Aufwandsgründen nur $n = 30$ verwenden, so gibt es $\binom{300}{30} \approx 1,7 \cdot 10^{41}$ verschiedene Untermengen mit 30 Merkmalen. Abgesehen von einigen einfachen Spezialfällen mit sehr kleinen Werten für n' und n wird es also schwierig sein, die optimale Untermenge zu bestimmen. Daher muss man nach Festlegung eines Gütemaßes für Merkmale auch noch ein *Auswahlverfahren* festlegen, mit dem man eine möglichst geeignete Untermenge mit erträglichem Aufwand finden kann.

Natürlich kann man statistische Abhängigkeiten zwischen Merkmalen zur Vereinfachung vernachlässigen und als beste Untermenge mit n Merkmalen die n am besten bewerteten wählen; tatsächlich wird häufig so verfahren. Man kann aber Beispiele dafür konstruieren, dass selbst bei klassenweise statistisch unabhängigen Merkmalen dieses Verfahren nicht immer optimal ist. Bewertet man jedes der n' Merkmale einzeln für sich und wählt die n einzeln am besten bewerteten aus, so ist das nicht notwendig die beste Untermenge mit n Merkmalen.

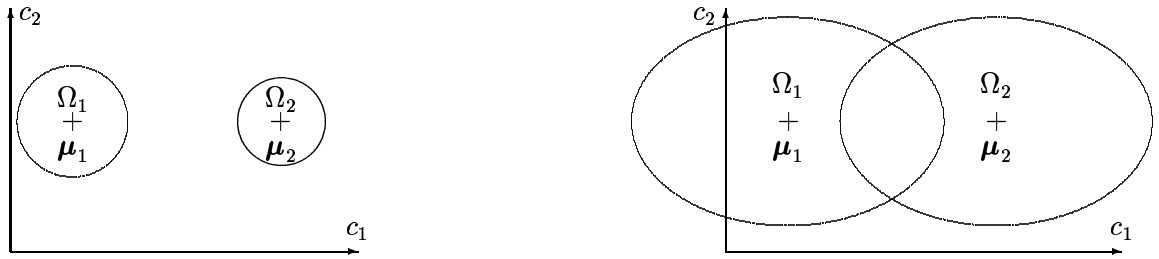


Bild 3.9.1: Beispiele für Bereiche, die von Merkmalen verschiedener Klassen eingenommen werden. Der Abstand der Mittelwerte $|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|$ ist allein nicht ausreichend, die Güte der Merkmale zu beurteilen.

Aus der obigen Diskussion geht hervor, dass es i. Allg. nicht möglich ist, die beste Untermenge von Merkmalen zu bestimmen. Andererseits liefern erfahrungsgemäß auch einfache Auswahlverfahren bereits wesentlich bessere Ergebnisse als eine Zufallsauswahl. Es wird noch erwähnt, dass oft auch die Verfahren von Abschnitt 3.8 als Merkmalsauswahl bezeichnet werden, da eine Reduzierung der Zahl der Variablen erreicht wird. Der Unterschied ist, dass dort neue Merkmale durch Linearkombination der vorhandenen gebildet werden, während hier die n besonders geeigneten unverändert aus der Menge der vorgegebenen übernommen werden.

3.9.2 Gütemaße für Merkmale

Der erste Schritt zur Auswahl einer Untermenge von Merkmalen aus einer Menge vorgegebener Merkmale ist, wie im vorigen Abschnitt erörtert, die Vorgabe eines Maßes zur Bewertung der Güte von Merkmalen. Dieses Gütemaß sollte im Zusammenhang mit der Fehlerwahrscheinlichkeit bei der Klassifikation stehen. Theoretisch besonders befriedigend sind natürlich solche Gütemaße, mit denen sich sehr enge obere und untere Schranken der Fehlerwahrscheinlichkeit angeben lassen. Das Gütemaß sollte aber auch numerisch noch mit vertretbarem Aufwand berechenbar sein, um für Zwecke der Musterklassifikation praktisch interessant zu sein; besonders günstig sind dafür solche Gütemaße, für die sich bei bestimmten Verteilungsdichten der Merkmale geschlossene Formeln angeben lassen. Diese beiden sich widersprechenden Forderungen führen zu der Vermutung, dass eine enge Abschätzung der Fehlerwahrscheinlichkeit (im Extremfall die Fehlerwahrscheinlichkeit selbst) numerisch nicht mehr auswertbar ist und eine auswertbare Abschätzung nur sehr grob ist. Ein sinnvoller Kompromiss wird stets vom jeweiligen Problem und der verfügbaren Rechenkapazität abhängen.

Zunächst geht aus Bild 3.9.1 hervor, dass die Güte von Merkmalen sicher mit dem Abstand von Merkmalsvektoren verschiedener Klassen zusammenhängt, dass aber der Abstand der Mittelwerte allein nicht ausreicht, um vernünftige Aussagen zu bekommen. Im Allgemeinen kommt es darauf an, ein geeignetes Maß für den *Abstand der Verteilungsdichten* der Merkmale aus verschiedenen Klassen zu finden, und dabei spielen *alle* Parameter eine Rolle. Dieses wird auch in Abschnitt 4.8.4 in (4.8.26), S. 424, und (4.8.44), S. 427, wieder aufgegriffen. Praktisch alle Gütemaße für Merkmale beruhen daher auf geeigneten verallgemeinerten Abstandsmaßen. In Abschnitt 4.1.4 wird gezeigt, dass der Klassifikator, der die Fehlerwahrscheinlichkeit p_f minimiert (der sog. BAYES-Klassifikator, Satz 4.3), die a posteriori Wahrscheinlichkeiten $p(\Omega_\kappa|\boldsymbol{c}),\ \kappa = 1, \ldots, k$ der Klassen berechnet und sich für die Klasse mit maximaler a posteriori Wahrscheinlichkeit entscheidet. Die Fehlerwahrscheinlichkeit dieses Klassifikators ist p_B.

Mit dem Vektor der a posteriori Wahrscheinlichkeiten wird der sog. BAYES-**Abstand**

$$B = \int_{R_{\boldsymbol{c}}} \sum_{\kappa=1}^{k} p^2(\Omega_\kappa|\,\boldsymbol{c}\,) p(\boldsymbol{c}\,)\,\mathrm{d}\boldsymbol{c} \tag{3.9.1}$$

definiert. Dabei ist

$$p(\boldsymbol{c}\,) = \sum_{\kappa=1}^{k} p_\kappa p(\boldsymbol{c}|\Omega_\kappa) \tag{3.9.2}$$

die Verteilungsdichte der Merkmalsvektoren. Der BAYES-Abstand ist also der *Erwartungswert* des Betragsquadrates des Vektors

$$\boldsymbol{\pi} = (p(\Omega_1|\,\boldsymbol{c}\,), \ldots, p(\Omega_\kappa|\,\boldsymbol{c}\,))^{\mathsf{T}}\ . \tag{3.9.3}$$

Ein großer Wert von B bedeutet, dass im Mittel eine sichere Klassifikation möglich ist, dass also die Merkmale geeignet sind. Die kleinstmögliche Fehlerwahrscheinlichkeit p_B erreicht der oben erwähnte Klassifikator, und es gilt die Abschätzung

$$\frac{1-B}{2} \le 1-\sqrt{B} \le \frac{k-1}{k}\left(1-\sqrt{\frac{kB-1}{k-1}}\right) \le p_B \le 1-B\ . \tag{3.9.4}$$

Ist p_B klein, etwa $p_B \simeq 0,1$, so werden mit guter Näherung die drei unteren Schranken für p_B gleich, und es gilt die besonders einfache Abschätzung

$$\boxed{\frac{1-B}{2} \le p_B \le 1-B\ .} \tag{3.9.5}$$

Ein anderes Abstandsmaß ist die **bedingte Entropie** oder *Equivokation*

$$H = \int_{R_{\boldsymbol{c}}} \left(-\sum_{\kappa=1}^{k} p_\kappa\, p(\boldsymbol{c}|\Omega_\kappa) \ln p(\Omega_\kappa|\,\boldsymbol{c}\,)\right) \mathrm{d}\boldsymbol{c}\ . \tag{3.9.6}$$

Für diese erhält man die Abschätzung

$$\boxed{p_B \le (1-B) \le \frac{H}{2}\ .} \tag{3.9.7}$$

Die Maße in (3.9.1), (3.9.6) haben den Vorteil, dass sie direkt den *allgemeinen Fall* von k Musterklassen erfassen. Aus (3.9.7) geht hervor, dass man mit dem BAYES-Abstand i. Allg. eine bessere Abschätzung erhält als mit der Equivokation. Es wurde sogar die Vermutung geäußert, dass man wahrscheinlich keine besseren Abschätzungen als die in (3.9.7) wird finden können.

Die obigen engen Abschätzungen haben den Nachteil, dass sie numerisch in geschlossener Form nicht auswertbar sind. Es ist im Prinzip möglich, den BAYES-Abstand B oder die Equivokation H mit einer Stichprobe von Mustern zu schätzen. Ein Schätzwert $\widehat{B}$ für B ist z. B.

$$\widehat{B} = \frac{1}{N}\sum_{\varrho=1}^{N}\sum_{\kappa=1}^{k} p^2(\Omega_\kappa|\,{}^{\varrho}\boldsymbol{c}\,)\ . \tag{3.9.8}$$

Natürlich ist das wenig sinnvoll, da man auch in (3.9.8) die a posteriori Wahrscheinlichkeiten berechnen muss, und wenn man diese hat, kann man genauso gut den Merkmalsvektor ${}^{\varrho}\boldsymbol{c}$ aus der bekannten Stichprobe klassifizieren und direkt die **Fehlerrate** $\widehat{p}_B$, d. h. einen *Schätzwert* der minimalen Fehlerwahrscheinlichkeit p_B gemäß

$$\widehat{p}_B = \frac{\text{Zahl der mit dem opt. Klass. } \textit{falsch} \text{ klassif. Muster}}{\text{Gesamtzahl der klassifizierten Muster}} \tag{3.9.9}$$

berechnen. In Kapitel 4 wird diskutiert, dass die Berechnung der a posteriori Wahrscheinlichkeiten wegen der dafür erforderlichen bedingten Dichten $p(\boldsymbol{c}|\Omega_\kappa)$ der Merkmalsvektoren i. Allg. nur näherungsweise möglich ist. Mit dem ebenfalls in Kapitel 4 behandelten Nächster–Nachbar–Klassifikator (NN–Klassifikator) ist es möglich, nichtparametrische Schätzwerte der Fehlerwahrscheinlichkeit *ohne* Kenntnis der Dichten $p(\boldsymbol{c}|\Omega_\kappa)$ zu berechnen. Bezeichnet man die Fehlerwahrscheinlichkeit des NN–Klassifikators mit p_N, so ist ein Schätzwert $\widehat{p}_N$ gegeben durch

$$\widehat{p}_N = \frac{\text{Zahl der mit dem NN–Klassifikator } \textit{falsch} \text{ klassif. Muster}}{\text{Gesamtzahl der klassifizierten Muster}} \; . \tag{3.9.10}$$

Zudem ist bekannt, dass p_N höchstens doppelt so groß wie p_B ist, sodass man mit $\widehat{p}_N$ auch eine Abschätzung für $\widehat{p}_B$ erhält. Ein besserer Schätzwert ergibt sich, wenn man den mNN–Klassifikator verwendet. Entsprechend (3.9.9), (3.9.10) lassen sich auch für irgendwelche anderen Klassifikatoren Fehlerwahrscheinlichkeiten schätzen. Diese Schätzwerte sind **Gütemaße** für Merkmale, deren Berechnung zwar aufwendig, aber mit einem modernen Großrechner ohne weiteres möglich ist. Es ist aber zu beachten, dass für Zwecke der Merkmalsauswahl noch eines der im nächsten Abschnitt erörterten suboptimalen **Auswahlverfahren** anzuschließen ist; dieses erfordert in der Regel die wiederholte Auswertung von (3.9.9) oder (3.9.10). Dadurch können die Gütemaße $\widehat{p}_B$ oder $\widehat{p}_N$ auch auf Großrechnern zu untragbaren Rechenzeiten führen, und daher ist es sinnvoll, nach einfacheren Gütemaßen zu suchen. Es wird noch erwähnt, dass sich bei großer a priori Wahrscheinlichkeit einer Klasse Beispiele konstruieren lassen, in denen die über alle Klassen gemittelte Fehlerwahrscheinlichkeit zur Auswahl trennscharfer Merkmale ungeeignet ist. Man muss dann entweder andere Gütemaße verwenden oder die Auswahlverfahren 3 oder 4 in Abschnitt 3.9.3, da diese auf klassenbedingten Gütemaßen basieren.

Es gibt verschiedene Vorschläge für Gütemaße $G_{\kappa\lambda}$, die sich nur auf die Unterscheidung zweier Klassen Ω_κ und Ω_λ beziehen. Bezeichnet man mit $p_{B_{\kappa\lambda}}$ die mit dem optimalen Klassifikator erreichbare (minimale) Fehlerwahrscheinlichkeit, so gibt es oft auch Abschätzungen von $p_{B_{\kappa\lambda}}$ mit dem Gütemaß $G_{\kappa\lambda}$. In der Regel wird man aber $k > 2$ Klassen haben, sodass man $G_{\kappa\lambda}$ für diesen Fall verwenden muss. Eine Verallgemeinerung gibt der Mittelwert

$$G = \frac{2}{k(k-1)} \sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-1} G_{\kappa\lambda} \; , \tag{3.9.11}$$

wobei $k(k-1)/2 = \binom{k}{2}$ die Zahl der *verschiedenen* Klassenpaare ist. Eine Verallgemeinerung der Abschätzung der Fehlerwahrscheinlichkeit ist praktisch nur über die Gleichung

$$p_B \leq \sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-1} p_{B_{\kappa\lambda}} \tag{3.9.12}$$

möglich. Ist

$$p_{B_{\kappa\lambda}} \leq \varphi(G_{\kappa\lambda}) \tag{3.9.13}$$

eine Abschätzung der paarweisen Fehlerwahrscheinlichkeit, so ergibt (3.9.13) eingesetzt in (3.9.12) eine Abschätzung von p_B mit Hilfe von $G_{\kappa\lambda}$. Allerdings sind Abschätzungen von p_B auf dieser Basis relativ grob, da schon (3.9.12) recht grob ist, besonders für eine große Klassenzahl k. Dafür haben einige der Maße $G_{\kappa\lambda}$ den Vorteil, dass sie numerisch mit relativ geringem Aufwand berechenbar sind.

Drei wichtige Gütemaße $G_{\kappa\lambda}$ sind:

1. Der BHATTACHARYYA-**Abstand**

$$\boxed{G^{\mathrm{B}}_{\kappa\lambda} = -\ln\left(\int \sqrt{p(\boldsymbol{c}|\Omega_\kappa)p(\boldsymbol{c}|\Omega_\lambda)}\,\mathrm{d}\boldsymbol{c}\right)\ .} \tag{3.9.14}$$

2. Die **Divergenz**

$$\boxed{G^{\mathrm{D}}_{\kappa\lambda} = \int (\ln p(\boldsymbol{c}|\Omega_\kappa) - \ln p(\boldsymbol{c}|\Omega_\lambda))(p(\boldsymbol{c}|\Omega_\kappa) - p(\boldsymbol{c}|\Omega_\lambda))\,\mathrm{d}\boldsymbol{c}\ .} \tag{3.9.15}$$

3. Die *Transinformation* bzw. **wechselseitige Information** (“mutual information”)

$$\boxed{\begin{aligned} G^{\mathrm{T}} &= -\sum_{\kappa=1}^{k}\int p_\kappa p(\boldsymbol{c}|\Omega_\kappa)\ln\frac{p(\boldsymbol{c}|\Omega_\kappa)}{p(\boldsymbol{c})}\,\mathrm{d}\boldsymbol{c} \\ &= -\sum_{\kappa=1}^{k}\int p(\boldsymbol{c},\Omega_\kappa)\ln\frac{p(\boldsymbol{c},\Omega_\kappa)}{p(\boldsymbol{c})p_\kappa}\,\mathrm{d}\boldsymbol{c}\ . \end{aligned}} \tag{3.9.16}$$

Darüberhinaus gibt es in der Literatur weitere Vorschläge, von denen einige als Beispiele erwähnt werden.

4. Der KOLMOGOROW-**Abstand**

$$G^{\mathrm{K}}_{\kappa\lambda} = \int |\,p(\Omega_\kappa\,|\,\boldsymbol{c}) - p(\Omega_\lambda\,|\,\boldsymbol{c})\,|p(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\ . \tag{3.9.17}$$

5. Der LISSAK–FU-**Abstand** (Verallgemeinerung von (3.9.17))

$$G^{\mathrm{L}}_{\kappa\lambda} = \int |p(\Omega_\kappa\,|\,\boldsymbol{c}) - p(\Omega_\lambda\,|\,\boldsymbol{c})|^\beta p(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\ . \tag{3.9.18}$$

6. Der CHERNOFF-**Abstand** (Verallgemeinerung von (3.9.14))

$$G^{\mathrm{C}}_{\kappa\lambda} = -ln\left(\int p(\boldsymbol{c}|\Omega_\kappa)^\alpha\, p(\boldsymbol{c}|\Omega_\lambda)^{1-\alpha}\,\mathrm{d}\boldsymbol{c}\right)\ , \qquad 0<\alpha<1\ . \tag{3.9.19}$$

7. Der MATUSITA-**Abstand**

$$G^{\mathrm{M}}_{\kappa\lambda} = \sqrt{\int\left(\sqrt{p(\boldsymbol{c}|\Omega_\kappa)} - \sqrt{p(\boldsymbol{c}|\Omega_\lambda)}\right)^2\,\mathrm{d}\boldsymbol{c}}\ . \tag{3.9.20}$$

8. Der PATRICK–FISHER-**Abstand**

$$G^{\mathrm{P}}_{\kappa\lambda} = \sqrt{\int (p_\kappa p(\boldsymbol{c}|\Omega_\kappa) - p_\lambda p(\boldsymbol{c}|\Omega_\lambda))^2 \,\mathrm{d}\boldsymbol{c}} \,. \tag{3.9.21}$$

9. Der **quadratische Abstand** (Spezialfall von (3.9.21))

$$G^{\mathrm{Q}}_{\kappa\lambda} = \int (p(\boldsymbol{c}|\Omega_\kappa) - p(\boldsymbol{c}|\Omega_\lambda))^2 \,\mathrm{d}\boldsymbol{c} \,. \tag{3.9.22}$$

Alle Integrale sind oben als bestimmte Integrale über den gesamten n–dimensionalen Merkmalsraum $\mathbb{R}_{\boldsymbol{c}}$ zu verstehen. Obwohl die Liste der Gütemaße nicht vollständig ist, mag sie hier genügen. Allen Maßen, mit Ausnahme der Transinformation (3.9.16), ist gemeinsam, dass jeweils *ein Paar* von Klassen betrachtet wird (und nicht k Klassen *gemeinsam*). Die Maße $G_{\kappa\lambda}$ nehmen kleine Werte an für $p(\boldsymbol{c}|\Omega_\kappa) = p(\boldsymbol{c}|\Omega_\lambda)$ und $p_\kappa = p_\lambda$, und sie nehmen große Werte an, wenn $p(\boldsymbol{c}|\Omega_\kappa) = 0$ für $p(\boldsymbol{c}|\Omega_\lambda) \neq 0$. Diese Eigenschaft ist nützlich, da im ersten Falle die Merkmale zur Unterscheidung der Klassen ungeeignet sind, im zweiten Falle gestatten sie eine vollkommene Unterscheidung. Es handelt sich bei den $G_{\kappa\lambda}$ um Größen, die den „Abstand" zwischen den bedingten Dichten $p(\boldsymbol{c}|\Omega_\kappa)$ und $p(\boldsymbol{c}|\Omega_\lambda)$ messen, und je größer dieser Abstand, desto besser die Merkmale. Diese Idee wird im BHATTACHARYYA-Abstand $G^{\mathrm{B}}_{\kappa\lambda}$ in (3.9.14) direkt umgesetzt.

Sowohl die Divergenz $G^{\mathrm{D}}_{\kappa\lambda}$ in (3.9.15) als auch die Transinformation G^{T} in (3.9.16) gehen auf informationstheoretische Begriffe zurück, die hier kurz ohne weitere Beweise angeführt werden. Die **Entropie** $H(\Omega)$ der Klassen Ω_κ ist definiert durch

$$H(\Omega) = -\sum_\kappa p_\kappa \log_2 p_\kappa \,. \tag{3.9.23}$$

Sie ist ein Maß für die Unsicherheit über das Auftreten von Werten von Ω, bzw. ein Maß für die *Information*, die man gewinnt, wenn ein Wert von Ω (eine Klasse) beobachtet wird. Die **bedingte Entropie** $H(\Omega|\boldsymbol{c})$ wurde bereits in (3.9.6) eingeführt und ist

$$H(\Omega|\boldsymbol{c}) = -\sum_\kappa \int_{\mathbb{R}_{\boldsymbol{c}}} p(\boldsymbol{c}, \Omega_\kappa) \log_2 p(\Omega_\kappa|\boldsymbol{c}) \,\mathrm{d}\boldsymbol{c} \,. \tag{3.9.24}$$

Die **Transinformation** (*wechselseitige Information*) $G^{\mathrm{T}}(\Omega;\boldsymbol{c})$ ist definiert durch

$$\begin{aligned} G^{\mathrm{T}}(\Omega;\boldsymbol{c}) &= G^{\mathrm{T}}(\boldsymbol{c};\Omega) = -\sum_\kappa \int_{\mathbb{R}_{\boldsymbol{c}}} p(\boldsymbol{c}, \Omega_\kappa) \log \frac{p(\boldsymbol{c}, \Omega_\kappa)}{p(\boldsymbol{c}) p_\kappa} \\ &= H(\boldsymbol{c}) - H(\boldsymbol{c}|\Omega) = H(\Omega) - H(\Omega|\boldsymbol{c}) \\ &= H(\boldsymbol{c}) + H(\Omega) - H(\boldsymbol{c}, \Omega) \,. \end{aligned} \tag{3.9.25}$$

Die *Minimierung* der bedingten Entropie entspricht also bei konstanten a priori Wahrscheinlichkeiten p_κ der *Maximierung* der wechselseitigen Information. Im Hinblick auf (3.9.26) lässt sich G^{T} auch als ein Maß für den *Abstand* der Dichten $p(\boldsymbol{c}, \Omega_\kappa)$ und $p(\boldsymbol{c})p_\kappa$ auffassen. Die **relative Entropie** oder der KULLBACK–LEIBLER-**Abstand** zwischen zwei Verteilungsdichten $p_1 = p(\boldsymbol{c}|\Omega_1), p_2 = p(\boldsymbol{c}|\Omega_2)$ ist definiert durch

$$D(p_1; p_2) = \int p_1 \ln \frac{p_1}{p_2} \,\mathrm{d}\boldsymbol{c} \,. \tag{3.9.26}$$

Da i. Allg. $D(p_1;p_2) \neq D(p_2;p_1)$ ist, kann man den Abstand durch $D(p_1,p_2) = D(p_1;p_2) + D(p_2;p_1)$ symmetrisieren. Das ist offenbar gerade die in (3.9.15) eingeführte Divergenz.

Die Transinformation G^{T} in (3.9.16) ist ein Maß, das sich auf k Klassen, nicht nur auf ein Paar, bezieht. Sie wurde hier erwähnt und nicht im Zusammenhang mit (3.9.6), da sich aus G^{T} eine ganze Klasse weiterer Abstandsmaße ergibt. Die Transinformation ist nämlich ein Maß dafür, welche Information die Beobachtung eines Merkmalsvektors $\boldsymbol{c}$ über die Klasse Ω_κ liefert. Sind $\boldsymbol{c}$ und Ω_κ statistisch unabhängig, d. h. ist

$$p(\boldsymbol{c},\Omega_\kappa) = p(\boldsymbol{c})p(\Omega_\kappa) = p(\boldsymbol{c})p_\kappa \ , \tag{3.9.27}$$

so enthält die Beobachtung von $\boldsymbol{c}$ offensichtlich *keine* Information über Ω_κ, und es ist $G^{\mathrm{T}} = 0$. Der Maximalwert

$$G^{\mathrm{T}}_{max} = -\sum_{\kappa=1}^{k} p_\kappa \log p_\kappa \tag{3.9.28}$$

wird angenommen, wenn $\boldsymbol{c}$ die Klasse *eindeutig* bestimmt. Der Informationsgewinn entspricht dann der *Entropie* (3.9.23) der Klassen. Wie erwähnt kann man G^{T} als Maß für den „Abstand" der Dichten $p(\boldsymbol{c},\Omega_\kappa)$ und $p(\boldsymbol{c})p_\kappa$ auffassen. Andere Abstandsmaße für diese beiden Dichten erhält man, wenn man $p(\boldsymbol{c}|\Omega_\kappa)$ bzw. $p(\boldsymbol{c}|\Omega_\lambda)$ in den obigen Maßen $G_{\kappa\lambda}$ durch $p(\boldsymbol{c},\Omega_\kappa)$ bzw. $p(\boldsymbol{c})p_\kappa$ ersetzt. Zur Berechnung von Schätzwerten der Transinformation mit einer PARZEN-Schätzung (s. Abschnitt 4.2.6) wird auf die Literatur verwiesen.

Die praktische Bedeutung obiger Gütemaße liegt, wie erwähnt, darin, dass sich für bestimmte Fälle *geschlossene* Lösungen der Integrale angeben lassen und dass für einige der Maße *Abschätzungen der Fehlerwahrscheinlichkeit* bekannt sind. Beispielsweise gilt für $G^{\mathrm{L}}_{\kappa\lambda}$ die Abschätzung

$$\frac{1}{2}\left(1-(G^{\mathrm{L}}_{\kappa\lambda})^{1/\beta}\right) \leq p_{B_{\kappa\lambda}} \leq \frac{1}{2}\left(1-G^{\mathrm{L}}_{\kappa\lambda}\right) \tag{3.9.29}$$

und für $G^{\mathrm{K}}_{\kappa\lambda}$ gilt exakt

$$p_{B_{\kappa\lambda}} = \frac{1}{2}\left(1-G^{\mathrm{K}}_{\kappa\lambda}\right) \ . \tag{3.9.30}$$

Dabei ist $p_{B_{\kappa\lambda}}$ wie in (3.9.13) die bei der Unterscheidung von $\Omega\kappa$ und Ω_λ minimal erreichbare Fehlerwahrscheinlichkeit. Weiterhin gilt

$$p_{B_{\kappa\lambda}} \leq \sqrt{p_\kappa p_\lambda}\exp\left[-G^{\mathrm{B}}_{\kappa\lambda}\right] \qquad . \tag{3.9.31}$$

Diese Abschätzungen bestätigen die intuitive Einsicht, dass Merkmale mit großen Werten von $G_{\kappa\lambda}$ gut sind, da die obere Schranke der Fehlerwahrscheinlichkeit umso kleiner wird je größer $G_{\kappa\lambda}$ ist. Wenn man annimmt, dass die Merkmalsvektoren *klassenweise normalverteilt* sind, dass also

$$p(\boldsymbol{c}|\Omega_\kappa) = \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}_\kappa|}}\exp\left[-\frac{1}{2}(\boldsymbol{c}-\boldsymbol{\mu}_\kappa)^{\mathsf{T}}\boldsymbol{\Sigma}_\kappa^{-1}(\boldsymbol{c}-\boldsymbol{\mu}_\kappa)\right] \tag{3.9.32}$$

ist, so lassen sich für den BHATTACHARYYA-Abstand und die Divergenz geschlossene Formeln angeben. Es gilt

$$G^{\mathrm{B}}_{\kappa\lambda} \ = \ \frac{1}{8}(\boldsymbol{\mu}_\kappa-\boldsymbol{\mu}_\lambda)^{\mathsf{T}}\left(\frac{\boldsymbol{\Sigma}_\kappa+\boldsymbol{\Sigma}_\lambda}{2}\right)^{-1}(\boldsymbol{\mu}_\kappa-\boldsymbol{\mu}_\lambda) + \frac{1}{2}\ln\left[\frac{|(\boldsymbol{\Sigma}_\kappa+\boldsymbol{\Sigma}_\lambda)|}{2\sqrt{|\boldsymbol{\Sigma}_\kappa||\boldsymbol{\Sigma}_\lambda|}}\right] \ , \tag{3.9.33}$$

$$G^{\mathrm{D}}_{\kappa\lambda} = \frac{1}{2}(\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda)^{\mathsf{T}}(\boldsymbol{\Sigma}_\kappa^{-1} + \boldsymbol{\Sigma}_\lambda^{-1})(\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda) + \frac{1}{2}\mathrm{Sp}\left(\boldsymbol{\Sigma}_\kappa^{-1}\boldsymbol{\Sigma}_\lambda + \boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{\Sigma}_\kappa - 2\boldsymbol{I}\right) \ . \tag{3.9.34}$$

Ist $\boldsymbol{\Sigma}_\kappa = \boldsymbol{\Sigma}_\lambda = \boldsymbol{\Sigma}$, so erhält man

$$\boxed{G^{\mathrm{A}}_{\kappa\lambda} = (\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_\lambda) = G^{\mathrm{D}}_{\kappa\lambda} = 8G^{\mathrm{B}}_{\kappa\lambda} \ ;} \tag{3.9.35}$$

diese Größe wird als MAHALANOBIS-**Abstand** bezeichnet (man vergleiche mit (3.8.79)). Sie wird häufig als numerisch einfach berechenbare Größe verwendet, auch wenn $\boldsymbol{\Sigma}_\kappa \neq \boldsymbol{\Sigma}_\lambda$ ist. Man setzt dann für $\boldsymbol{\Sigma}$ in (3.9.35)

$$\boldsymbol{\Sigma} = \frac{N_\kappa}{N}\boldsymbol{\Sigma}_\kappa + \frac{N_\lambda}{N}\boldsymbol{\Sigma}_\lambda \qquad \text{oder auch} \tag{3.9.36}$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{j=1}^{N} {}^j\boldsymbol{c}\,{}^j\boldsymbol{c}^{\mathsf{T}} - \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}} \ , \tag{3.9.37}$$

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{j=1}^{N} {}^j\boldsymbol{c} \ .$$

Ein besonders einfaches Maß ergibt sich, wenn man *nur ein* Merkmal c_ν betrachtet und $N_\kappa/N = N_\lambda/N = 1/2$ ist. In diesem Falle reduziert sich (3.9.35) mit (3.9.36) auf

$$G^{\mathrm{A}}_{\kappa\lambda\nu} = \frac{2(\mu_{\kappa\nu} - \mu_{\lambda\nu})^2}{\sigma^2_{\kappa\nu} + \sigma^2_{\lambda\nu}} \tag{3.9.38}$$

oder mit (3.9.38) auf

$$G^{\mathrm{A}}_{\kappa\lambda\nu} = \frac{(\mu_{\kappa\nu} - \mu_{\lambda\nu})^2}{\sigma^2} \ . \tag{3.9.39}$$

Damit steht eine Reihe zunehmend spezialisierter Gütemaße zur Verfügung, von denen (3.9.38), (3.9.39) am einfachsten zu berechnen sind. Allerdings wird damit jedes Merkmal für sich bewertet, *ohne* Berücksichtigung statistischer Abhängigkeiten zu anderen Merkmalen. Dazu kommt, dass nur Momente bis zur zweiten Ordnung in (3.9.38) berücksichtigt werden, d. h. die Merkmale müssen zumindest näherungsweise normalverteilt sein, bzw. wenigstens eine unimodale Verteilung besitzen. Mit (3.9.33) – (3.9.35) werden *lineare* statistische Abhängigkeiten zwischen den n Merkmalen des Merkmalsvektors $\boldsymbol{c}$ erfasst. Hierbei ist Voraussetzung, dass die Merkmalsvektoren näherungsweise normalverteilt sind, damit diese drei Maße eine zuverlässige Bewertung ergeben. Wenn man *allgemeine* statistische Abhängigkeiten mit berücksichtigen will – oder, was dasselbe ist, Momente von höherer als zweiter Ordnung – so bleiben nur die allgemeinen, auf Verteilungsdichten beruhenden Gütemaße (3.9.14) – (3.9.22). Zu ihrer Berechnung sind i. Allg. die n–dimensionalen Verteilungsdichten $p(\boldsymbol{c}|\Omega_\kappa)$ zu schätzen. Für praktisch interessante Werte von n, etwa $n = 10$ bis $n = 100$, und nicht normalverteilte Merkmalsvektoren gibt es dafür keine numerisch auswertbaren Verfahren. Das bedeutet, dass man entweder die tatsächliche Verteilungsdichte durch eine Normalverteilungsdichte approximiert oder jedes Merkmal für sich bewertet, also nur mit eindimensionalen Dichten arbeitet. Solche eindimensionalen Dichten kann man relativ einfach durch das Histogramm, also durch Abzählen relativer

Häufigkeiten analog zu Abschnitt 2.2.2, approximieren. Zur Bestimmung von Verteilungsdichten wird auch auf Abschnitt 4.2 verwiesen.

Die Beschränkung auf einzelne Merkmale c_ν und die Schätzung der Verteilungsdichten mit dem Histogramm hat den weiteren Vorteil, dass sich die Integrale in (3.9.14) – (3.9.22) dann auf einfach auszuwertende Summen reduzieren. Als Beispiel wird hier nur die Transinformation G^{T} in (3.9.16) betrachtet. Das Merkmal c_ν möge jeweils einen von m möglichen diskreten Werten $c_{\nu j},\ j = 1,\ldots,m$ annehmen. Wie in Abschnitt 2.1.3 erörtert wurde, lässt sich ein kontinuierlicher Wertebereich für c_ν stets in dieser Weise quantisieren. Damit geht (3.9.16) über in die diskrete Form

$$G^{\mathrm{T}} = \sum_{\kappa=1}^{k}\sum_{j=1}^{m} p_\kappa p(c_{\nu j}\,|\,\Omega_\kappa)\log\frac{p(c_{\nu j}\,|\,\Omega_\kappa)}{p(c_{\nu j})}\ . \tag{3.9.40}$$

Wenn eine genügend große klassifizierte Stichprobe gegeben ist, bereitet die Schätzung von $p(c_{\nu j}\,|\,\Omega_\kappa)$ keine Probleme. Die obige Diskussion zeigt, dass man unter Umständen nur die Güte einzelner Merkmale c_ν für sich beurteilen wird. Sind aber c_μ und c_ν zwei Merkmale und gilt für eine reelle, nicht abnehmende Funktion g die Beziehung

$$p(c_\nu = g(c_\mu)) \;=\; 1\ , \tag{3.9.41}$$

dann kann man auf c_ν verzichten. Es lässt sich zeigen, dass es dann stets eine nur von c_μ abhängige Entscheidungsregel gibt, welche Muster mit gleicher Fehlerwahrscheinlichkeit wie eine von c_μ und c_ν abhängige Entscheidungsregel klassifiziert. Es ist also nützlich, nicht solche Merkmale zu verwenden, zwischen denen Abhängigkeiten bestehen. Mit dem **Korrelationskoeffizienten**

$$\rho_{\mu\nu} = \frac{E\{(c_\mu - E\{c_\mu\})(c_\nu - E\{c_\nu\})\}}{\sqrt{E\{(c_\mu - E\{c_\mu\})^2\}E\{(c_\nu - E\{c_\nu\})^2\}}} \tag{3.9.42}$$

lassen sich wenigstens *lineare* Abhängigkeiten der Form

$$c_\nu = ac_\mu + b \tag{3.9.43}$$

zwischen zwei Merkmalen bewerten. Natürlich werden solche auch vom MAHALANOBIS-Abstand (3.9.35) erfasst. Dieser ist also geeignet, eine *Menge* von Merkmalen bzw. einen Merkmalsvektor unter Berücksichtigung linearer statistischer Abhängigkeiten zu bewerten, wenn dieser, wie schon gesagt, näherungsweise normalverteilt ist.

3.9.3 Auswahlverfahren

Im Abschnitt 3.9.1 wurde dargelegt, dass zur exakten Bestimmung der besten Untermenge von n Merkmalen alle $\binom{n'}{n}$ Untermengen durch Schätzung der Fehlerrate zu bewerten sind und die mit der kleinsten auszuwählen ist. Diese Methode scheidet i. Allg. wegen des damit verbundenen Aufwandes aus. Bild 3.9.2a verdeutlicht die vollständige Suchmethode. Jeder Kantenzug von links nach rechts enthält die Bewertung einer Untermenge mit $n = 3$ Merkmalen, z. B. der dick gezeichnete die der Menge c_1, c_3, c_4. Entsprechend den $\binom{n'}{n}$ Untermengen gibt es $\binom{n'}{n}$ Kantenzüge. Das triviale Verfahren der zufälligen Auswahl von n aus n' Merkmalen scheidet wegen der i. Allg. unbefriedigenden Qualität der so gefundenen Untermenge aus. Zwischen diesen beiden Extremen liegen die im Folgenden diskutierten *Auswahlverfahren*.

Um eine Häufung von Indizes zu vermeiden, wird die Güte eines einzelnen Merkmals c_ν mit G_ν bezeichnet, die Güte einer Untermenge mit j Merkmalen mit G^j. Die Erkennungsrate $1 - \widehat{p}_B$ oder $1 - \widehat{p}_N$, mit $\widehat{p}_B$, $\widehat{p}_N$ aus (3.9.9), (3.9.10), eignet sich sowohl zur Bewertung der Güte G_ν eines einzelnen Merkmals c_ν als auch zur Bewertung der Güte G^j einer Untermenge von j Merkmalen. Die Maße (3.9.14) – (3.9.22) sind zwar im Prinzip für die Bewertung einer Menge von Merkmalen geeignet, jedoch wegen der Problematik des Schätzens hochdimensionaler Verteilungsdichten praktisch auf die Bewertung einzelner Merkmale beschränkt; (3.9.38), (3.9.39) zusammen mit (3.9.11) eignen sich nur zur Bewertung einzelner Merkmale. Die Maße (3.9.33)–(3.9.35) sind auch vom Rechenaufwand her für eine Untermenge von j Merkmalen geeignet.

Einige mögliche Auswahlverfahren, kurz charakterisiert, sind:

1. Nimm die n *einzeln* am besten bewerteten Merkmale.
2. *Integriere* sukzessive jeweils das Merkmal, das relativ zu den schon vorhandenen *am besten* ist.
3. *Integriere* sukzessive jeweils das Merkmal, das den *größten* Beitrag zum *schwierigsten* Klassenpaar liefert.
4. *Eliminiere* sukzessive jeweils das Merkmal, das den *kleinsten* Beitrag zum *schwierigsten* Klassenpaar liefert.
5. Verwende ein *monotones* Gütemaß, um mit der branch–and–bound Suche eine im Sinne des Gütemaßes *optimale* Teilmenge zu finden.
6. Verwende ein *monotones* und *separierbares* Gütemaß, um mit der dynamischen Programmierung eine im Sinne des Gütemaßes *optimale* Teilmenge zu finden.
7. *Integriere* l gute Merkmale, dann *eliminiere* r schlechte Merkmale, bis die gewünschte Merkmalszahl erreicht ist, die sog. (l, r)–*Suche*.
8. Integriere und eliminiere Merkmale in Abhängigkeit von ihrem Beitrag zur Güte der Merkmalsmenge durch eine sog. *alternierende* Suche, die eine Variante der (l, r)–Suche ist.
9. Generiere mit genetischen Algorithmen Populationen (Teilmengen) von Merkmalen, um in mehreren Schritten eine besonders gute zu finden.

Die zufällige Auswahl von Merkmalen wurde oben als einfachste, aber schlechteste, Möglichkeit ausgelassen. Es werden nun zunächst einige Einzelheiten zu den ersten vier **heuristischen Auswahlverfahren** angegeben.

Dann folgen mit Nr. 5 bzw. 6 zwei systematische Verfahren auf der Basis einer "branch–and–bound" Suche bzw. der dynamischen Progammierung. Mit Nr. 7 und 8 werden zwei Varianten der (l, r)–Suche vorgestellt, die auch für nichtmonotone Gütekriterien geeignet sind. Genetische Algorithmen werden nur kurz skizziert.

1. Nimm die n *einzeln* am besten bewerteten Merkmale.

1.1 Man wähle eines der Maße G_ν zur Beurteilung der Güte eines einzelnen Merkmals c_ν aus Abschnitt 3.9.2.

1.2 Man berechne G_ν für alle n' vorgegebenen Merkmale $c_\nu,\ \nu = 1, \ldots, n'$.

1.3 Man wähle die n Merkmale mit den größten Werten von G_ν aus.

Bild 3.9.2b verdeutlicht dieses Auswahlverfahren. Jeder Kantenzug enthält die Bewertung eines einzelnen Merkmals. Mit $\max_1$ wird das am besten bewertete bezeichnet, mit $\max_2$ das am zweitbesten bewertete, usw. Dieses einfache Auswahlverfahren ist nach experimentellen Ergebnissen bereits deutlich besser als eine zufällige Auswahl.

2. *Integriere* sukzessive jeweils das Merkmal, das relativ zu den schon vorhandenen *am besten* ist.

2.1 Man wähle eines der Maße G^j zur Beurteilung der Güte einer Untermenge von j Merkmalen aus Abschnitt 3.9.2, z. B. (3.9.35).

2.2 Man berechne G^j für alle n' Merkmale allein, d. h. $j = 1$.

2.3 Als erstes Merkmal wird das mit dem größten Wert von G^j gewählt.

2.4 Es seien bereits $(j-1)$ Merkmale, $j \geq 2$, ausgewählt. Man berechne G^j für alle $(n'-j+1)$ Teilmengen mit j Merkmalen, wobei jede Teilmenge die schon ausgewählten $(j-1)$ Merkmale und ein weiteres enthält.

2.5 Als j–tes Merkmal wähle man das mit dem größten Wert von G^j aus.

2.6 Man wiederhole Schritt 2.4 und 2.5 bis n Merkmale ausgewählt sind.

Bild 3.9.2c1 zeigt dieses Auswahlverfahren. Jeder aus j Einzelkanten bestehende Kantenzug enthält die Bewertung einer Untermenge mit j Merkmalen. Der aus drei Einzelkanten bestehende, dick gezeichnete Kantenzug gehört z. B. zur Untermenge (c_5, c_1, c_3). Er muss das einzeln am besten bewertete Merkmal enthalten – das ist hier c_5 – sowie die beiden am besten bewerteten und c_5 enthaltenden Merkmale – das sind hier c_5 und c_1. In Bild 3.9.2c2 wurde nach Auswahl des ersten Merkmals die Darstellung so umgeordnet, dass das ausgewählte Merkmal zuunterst liegt. Nach Auswahl des zweiten Merkmals wird wieder so umgeordnet, dass dieses zu zweitunterst liegt, usw. Dieses Auswahlverfahren ist offensichtlich komplexer als das erste, liefert dafür auch bessere Ergebnisse.

3. *Integriere* sukzessive jeweils das Merkmal, das den *größten* Beitrag zum *schwierigsten* Klassenpaar liefert.

3.1 – 3.3 Wie 2.1 – 2.3, siehe oben.

3.4 Es seien bereits $(j-1)$ Merkmale, $j \geq 2$, ausgewählt. Man bestimme $G^{j-1}_{\kappa\lambda}$ für alle Klassenpaare und die schon ausgewählten Merkmale. Man ermittle das Klassenpaar mit dem kleinsten Wert von $G^{j-1}_{\kappa\lambda}$ und betrachte in Schritt 3.5 nur dieses Paar.

3.5 *Variante 1*: Man berechne für das in Schritt 3.4 ermittelte Klassenpaar den Wert von $G^j_{\kappa\lambda}$ für alle $(n'-j+1)$ Teilmengen mit j Merkmalen, wobei jede Teilmenge die schon ausgewählten $(j-1)$ Merkmale enthält. Als j–tes Merkmal wähle man das mit dem größten Wert von $G^j_{\kappa\lambda}$.

Variante 2: Man berechne $G_{\kappa\lambda\nu}$ für das in Schritt 3.4 ermittelte Klassenpaar und für alle $(n'-j+1)$ noch nicht ausgewählten Merkmale allein und wähle als j–tes Merkmal das mit dem größten Wert von $G_{\kappa\lambda\nu}$.

3.6 Man wiederhole Schritt 3.4 und 3.5 bis n Merkmale ausgewählt sind.

Dieses Verfahren vermeidet Probleme, die dadurch auftreten, dass bei Mittelung über alle Klassenpaare ein schwieriges Klassenpaar mit geringer a priori Wahrscheinlichkeit kaum eine Rolle spielt und daher Merkmale ausgewählt werden, mit denen dieses nur sehr schlecht unterschieden wird. Eine ähnliche Wirkung hat auch das nächste Auswahlverfahren.

4. *Eliminiere* sukzessive jeweils das Merkmal, das den *kleinsten* Beitrag zum *schwierigsten* Klassenpaar liefert.

4.1 Man berechne $G_{\kappa\lambda\nu}$ für jedes Klassenpaar und für alle n' Merkmale allein.

4.2 Es seien bereits l Merkmale, $l = 0, 1, \ldots, n' - n - 1$ eliminiert. Man berechne

$$G_{\kappa\lambda} = \sum_{\nu=1}^{n'-l} G_{\kappa\lambda\nu} \ . \tag{3.9.44}$$

4.3 Man bestimme das Klassenpaar mit dem kleinsten Wert von $G_{\kappa\lambda}$ und eliminiere das Merkmal, das zu diesem $G_{\kappa\lambda}$ den kleinsten Einzelbeitrag $G_{\kappa\lambda\nu}$ liefert.

4.4 Man wiederhole Schritt 4.2 und 4.3 bis von den anfänglichen n' Merkmalen nur noch n übrig sind.

Die Auswahlverfahren 1 und 4 sind rechnerisch am einfachsten, da jedes Merkmal nur für sich allein bewertet wird. Die Verfahren 2 und 3 berücksichtigen auch Beziehungen zu den schon ausgewählten Merkmalen. Im Verfahren 2 müssen insgesamt $n(n' - (n-1)/2)$ Untermengen mit einer von 1 bis n wachsenden Zahl von Merkmalen durchsucht werden. Für das Beispiel von Abschnitt 3.9.1 mit $n' = 300,\ n = 30$ bedeutet das statt $1,7 \cdot 10^{41}$ Untermengen „nur“ 8565. Allerdings wird man i. Allg. mit keinem dieser Auswahlverfahren die im Sinne von Definition 3.16 beste Untermenge finden. Abgesehen von der Zahl der zu durchsuchenden Untermengen hängt der erforderliche Aufwand auch vom verwendeten Gütekriterium ab, wie aus dem vorigen Abschnitt hervorgeht. Wählt man als Gütekriterium die mit dem NN–Klassifikator ermittelte Erkennungsrate $1 - \widehat{p}_N$ und speichert die erforderlichen Abstandsquadrate

$$d_{n-1}({}^j\boldsymbol{c}, {}^k\boldsymbol{c}) = \sum_{\nu=1}^{n-1} ({}^jc_\nu - {}^kc_\nu)^2 \tag{3.9.45}$$

in einer Abstandsmatrix, so lässt sich diese bei Hinzunahme eines weiteren n–ten Merkmals iterativ auffrischen gemäß

$$d_n({}^j\boldsymbol{c}, {}^j\boldsymbol{c}) = \sum_{\nu=1}^{n} ({}^jc_\nu - {}^kc_\nu)^2 = d_{n-1}({}^j\boldsymbol{c}, {}^k\boldsymbol{c}) + ({}^jc_n - {}^kc_n)^2 \ . \tag{3.9.46}$$

Damit wird der Aufwand für die wiederholte Berechnung von $\widehat{p}_N$ mit unterschiedlicher Zahl von Merkmalen wesentlich reduziert, jedoch muss man die Abstandsmatrix speichern. Die Auswahlverfahren 3 und 4 eignen sich dann, wenn die über alle Klassen gemittelten Gütemaße nicht genügend aussagekräftig sind, weil sehr starke Unterschiede in den Werten $G_{\kappa\lambda}$ für einzelne Klassenpaare auftreten.

Weitere heuristische Auswahlverfahren erhält man durch Vergrößerung der Zahl der durchsuchten Untermengen. In Bild 3.9.2d wird nicht nur das beste einzeln bewertete, sondern auch das am zweitbesten bewertete in die weitere Suche mit einbezogen. In Bild 3.9.2e wird zunächst die beste Untermenge mit zwei Merkmalen durch vollständige Suche über alle $\binom{n'}{2}$ Untermengen bestimmt, dann die beste Untermenge mit vier Merkmalen, welche die zwei schon ausgewählten enthält, usw. Es wurde wieder eine Umordnung analog Bild 3.9.2c2 angenommen.

Neben den obigen heuristischen Auswahlverfahren gibt es auch *systematische Auswahlverfahren* auf der Basis der dynamischen Programmierung und der „branch–and–bound“ Suche. Bei geeigneten Gütekriterien lässt sich mit der branch–and–bound Suche sogar die beste Untermenge finden, d. h. die beste im Sinne des Gütekriteriums, das i. Allg. *nicht* die Fehllerrate ist. Daher wird dieses Verfahren als nächstes und dann ein auf der dynamischen Programmierung beruhendes erläutert.

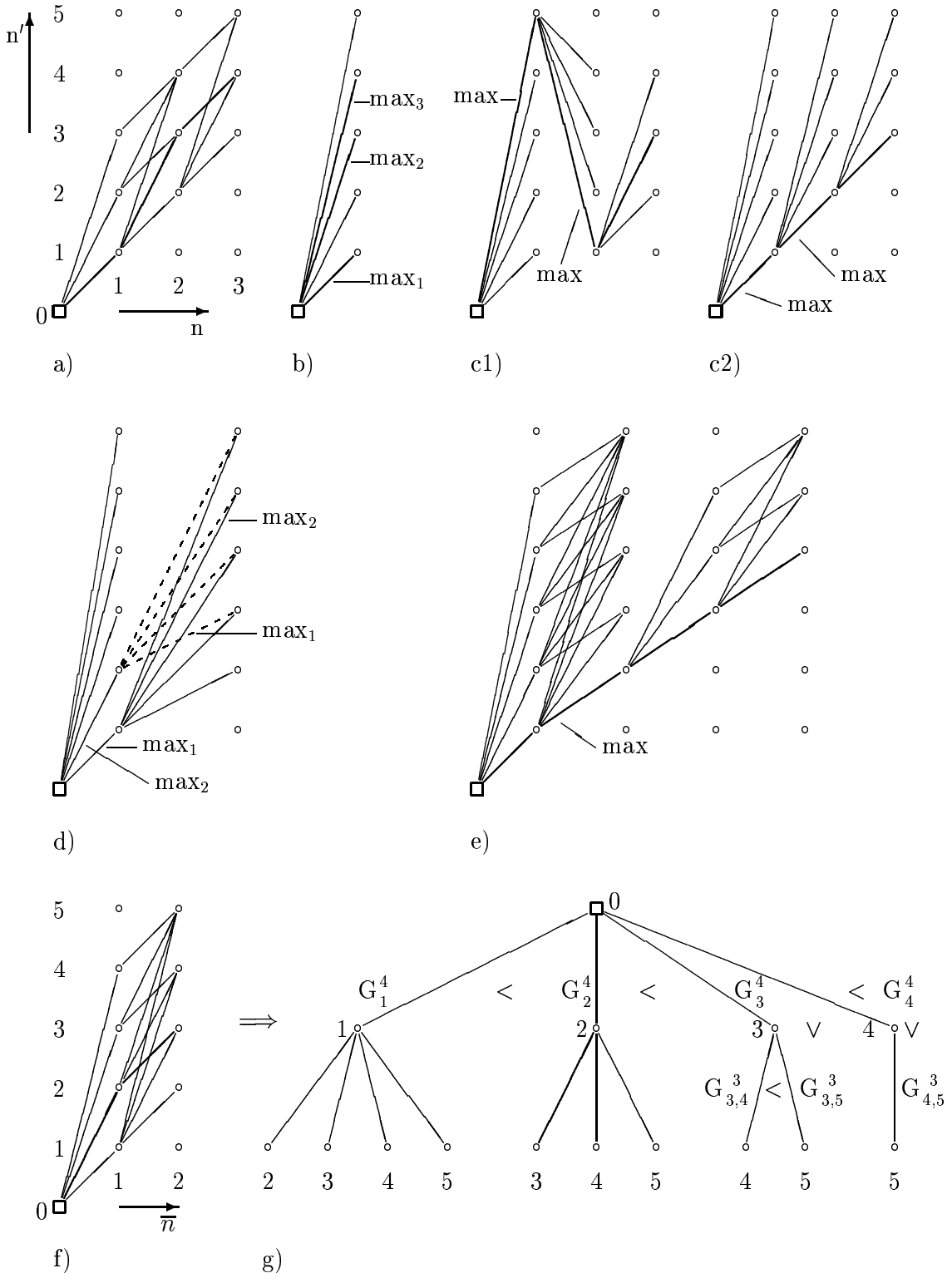


Bild 3.9.2: Auswahlverfahren für Merkmale. a) Alle *verschiedenen* Teilmengen mit 3 Merkmalen aus einer Gesamtmenge mit 5 Merkmalen sind durch Kanten repräsentiert; b) Auswahl der 3 besten Merkmale; c1) und c2) Auswahl des Merkmals, das relativ zu den schon vorhandenen am besten ist; d) und e) heuristische Erweiterungen der Suche; f) mögliche Eliminierungen von $\overline{n} = 2$ Merkmalen aus einer Gesamtmenge mit 5 Merkmalen; g) Anordnung von f) als Baum.

Bei der **„branch–and–bound" Suche** wird, wie in Bild 3.9.2f angedeutet ist, diejenige Untermenge mit $\overline{n} = n' - n$ Merkmalen bestimmt, deren *Komplement* die beste Untermenge mit n Merkmalen liefert, d. h. es werden $\overline{n}$ Merkmale *eliminiert*. Alle $\binom{n'}{\overline{n}} = \binom{n'}{n'-n} = \binom{n'}{n}$ Untermengen sind für $n' = 5,\ \overline{n} = 2$ wieder durch Kantenzüge dargestellt. Der dick gezeichnete Kantenzug gehört z. B. zur eliminierten Untermenge (c_2, c_3), deren Komplement die ausgewählte Merkmalsmenge (c_1, c_4, c_5) ergibt. Das Netzwerk von Bild 3.9.2f ist in Bild 3.9.2g als Baum gezeichnet. Jeder Untermenge entspricht ein Kantenzug von der Wurzel zu den Blättern. Diese Darstellung wird gewählt, weil die Suche durch eine Baumstruktur einfacher ist als durch einen allgemeinen Graphen.

Die Idee besteht darin, ein *monotones* Gütemaß zu verwenden. Damit ist gemeint, dass die Güte G^j einer Menge mit j Merkmalen *nicht kleiner* ist als die Güte G^{j-1} einer Menge, die aus diesen j Merkmalen *minus einem* besteht (s. (3.9.47)). Es ist bekannt, dass die Fehlerwahrscheinlichkeit diese Monotonieeigenschaft i. Allg. *nicht* besitzt, jedoch ist offensichtlich der MAHALANOBIS-Abstand (3.9.35) in diesem Sinne monoton, und das gilt auch für den BHATTACHARYYA-Abstand (3.9.14) und die Divergenz (3.9.15). Wir nehmen nun an, es sei eine Teilmenge $\{c_{\mu(1)}, \ldots, c_{\mu(\overline{n})}\}$ mit $\overline{n}$ Merkmalen bereits *eliminiert* worden. Die Güte der *verbleibenden* Merkmale, d. h. des *Komplements*, sei G. Als Alternative dazu werde die Elimination einer Teilmenge $\{c_{\nu(1)}, \ldots, c_{\nu(k)}\}$ mit $k < \overline{n}$ untersucht; die Güte der Komplementmenge sei G'. Da in der zweiten Teilmenge noch nicht $\overline{n}$ Merkmale eliminiert wurden, würde im nächsten Schritt ein weiteres Merkmal eliminiert werden; die Güte der nun resultierenden Komplementmenge sei G''. Wegen der Monotonie ist klar, dass dann $G' \geq G''$ ist. Wenn nun bereits $G \geq G'$ ist, dann ist von vornherein klar, dass wegen der Monotonie auch $G \geq G''$ ist. Das bedeutet, dass die Elimination weiterer Merkmale sich *erübrigt*, da die Güte der resultierenden Merkmalsmenge nur noch schlechter werden kann. Die weitere Suche durch Vergrößerung der Menge $\{c_{\nu(1)}, \ldots, c_{\nu(k)}\}$ kann abgebrochen werden und damit frühzeitig viele nutzlose Alternativen von der Suche ausgeschlossen werden. Die Effizienz der Suche wird dadurch gesteigert, dass man am Anfang bereits eine möglichst *hoch* bewertete Merkmalsmenge bereitstellt, weil damit viele Alternativen frühzeitig von der weiteren Suche ausgeschlossen werden. Diese Idee wird im Folgenden zusammengefasst.

5. branch–and–bound Suche zur Eliminierung der Untermenge $\overline{C}$ mit $\overline{n}$ Merkmalen, deren *Komplement* die beste Untermenge mit n Merkmalen ergibt.

5.1 Bezeichnung: $G^j_{\mu(1),\mu(2),\ldots}$ ist die Bewertung der Untermenge mit j Merkmalen, die man erhält, wenn man die Merkmale $c_{\mu(1)}, c_{\mu(2)}, \ldots$ eliminiert. Beispielsweise ist $G^3_{4,5}$ die Bewertung derjenigen Untermenge mit drei Merkmalen, die man erhält, wenn man aus der vorgegebenen Menge mit $n' = 5$ Merkmalen die Merkmale c_4 und c_5 eliminiert.

5.2 Man wähle ein *monotones* Gütemaß mit

$$G^{n'-1}_{\mu(1)} \geq G^{n'-2}_{\mu(1),\mu(2)} \geq \ldots \geq G^{n'-\overline{n}}_{\mu(1),\ldots,\mu(\overline{n})} = G^{n}_{\mu(1),\ldots,\mu(\overline{n})} \ . \tag{3.9.47}$$

Dieses gilt z. B. für den MAHALANOBIS-Abstand.

5.3 Ordne die Merkmale der ersten Ebene des Baumes so an, dass die Bewertungen von links nach rechts *zunehmen*. Die erste Ebene enthält $n+1$ Merkmale (Knoten). In Bild 3.9.2f,g ist angenommen, dass die Merkmale in diesem Sinne geordnet wurden.

5.4 Beginne mit dem am weitesten rechts liegenden Knoten, und gehe an ihm in $\overline{n}-1$ Schritten durch weitere Eliminierungen in die Tiefe. Das ergibt eine erste eliminierte Untermenge $\overline{C}$ mit $\overline{n}$ Merkmalen. Das *Komplement* von $\overline{C}$ ergibt eine Menge mit n Merkmalen; ihre Güte sei G.

5.5 Nimm in der betrachteten Ebene den am weitesten rechts liegenden noch nicht betrachteten Knoten als aktuellen Knoten. Wenn es in dieser Ebene keinen solchen Knoten mehr gibt, dann gehe zurück und suche die nächsthöhere Ebene im Baum mit mindestens einem Knoten, von dem noch nicht alle Nachfolger betrachtet wurden. Wenn es eine solche Ebene gibt, dann nimm den am weitesten rechts liegenden Knoten dieser Ebene als aktuellen Knoten, sonst ist der Algorithmus zu ENDE und der beste Wert des Gütemaßes ist G, die beste Untermenge das Komplement von $\overline{C}$.

5.6 Berechne den Wert des Gütemaßes im aktuellen Knoten, dieser Wert sei G'. Wenn $G' \leq G$, dann kann wegen (3.9.47) die diesen Knoten bzw. dieses Merkmal enthaltende Untermenge nicht optimal sein; der Knoten scheidet mit allen Nachfolgern von der Betrachtung aus; gehe zurück zu 5.5.

5.7 Wenn $G' > G$ und der aktuelle Knoten kein Blatt ist, dann bestimme alle seine Nachfolger. Ordne die Nachfolger analog Schritt 5.3. Nimm den am weitesten rechts liegenden Knoten als neuen aktuellen Knoten, gehe nach Schritt 5.6.

5.8 Wenn $G' > G$ und der betrachtete Knoten ein Blatt ist, dann setze $G = G'$, ersetze $\overline{C}$ durch die zu diesem Pfad gehörige Untermenge, gehe nach Schritt 5.5.

Dieser Algorithmus findet für jedes Gütemaß G, das (3.9.47) genügt, die beste Untermenge, da er alle Untermengen bewertet. Der wesentliche Schritt ist 5.6, da in diesem aussichtslose Untermengen frühzeitig ausgeschieden werden. Dadurch wird der Suchaufwand erheblich vermindert. Für kleinere Werte von n' und n, etwa $n' \simeq 24,\ n \simeq 12$ können damit die besten Merkmale gefunden werden.

Auswahlverfahren nach der **dynamischen Programmierung** (DP, s. Abschnitt 1.6.8) beruhen auf dem Optimalitätsprinzip, wonach eine optimale Strategie die Eigenschaft hat, dass unabhängig vom Anfangszustand und den Anfangsentscheidungen die folgenden Entscheidungen wieder eine optimale Strategie bilden; damit dieses gilt, müssen bestimmte Monotoniebedingungen erfüllt sein. Ein Auswahlalgorithmus ist der folgende:

6. Auswahl von n Merkmalen aus n' vorgegebenen mit der dynamischen Programmierung.

6.1 Man initialisiere n' Mengen $C_i^1 = \{c_i\},\ i = 1, \ldots, n'$, d. h. jede Menge enthält anfänglich ein Merkmal, und man wähle ein Bewertungsmaß für Merkmale, das monoton und separierbar ist; z. B. erfüllt der MAHALANOBIS-Abstand diese Bedingungen.

6.2 Für $j = 2, \ldots, n$ führe man Schritt 6.3 aus.

6.3 Bilde für ein bestimmtes i alle Mengen $C_{i\nu}^j = \{C_i^{j-1}, c_\nu | c_\nu \notin C_i^{j-1}\}$ und wähle als Menge C_i^j die am besten bewertete aus. Führe diesen Schritt für alle i von 1 bis n' aus.

6.4 Die n' Mengen C_i^n enthalten n Merkmale. Wähle als beste Menge mit n Merkmalen die am besten bewertete Menge aus C_i^n.

Bei dieser Vorgehensweise sind $n'(n'(n-1) - n(n-1)/2)$ Untermengen unterschiedlicher Größe zu durchsuchen. Mit $n' = 300$ und $n = 30$ ergibt das rund $2,5 \cdot 10^6$ Untermengen, sodass dieses Verfahren ebenfalls auf kleinere Merkmalszahlen beschränkt ist.

Die (l, r)**–Suche** gilt schließlich als ein Verfahren, das auch bei *nicht monotonen* Gütefunktionen – wie der Fehlerwahrscheinlichkeit – gute Ergebnisse liefert. Ihr Prinzip beruht darauf, zunächst die l besten Merkmale auszuwählen, davon dann die r schlechtesten Merkmale zu verwerfen und diesen Prozess zu wiederholen, bis n Merkmale vorliegen. Das Gütekriterium ist im Prinzip beliebig, insbesondere muss es, wie gesagt, nicht monoton sein (wie bei der branch–and–bound Suche), d. h. man kann auch die Fehlerwahrscheinlichkeit als Gütekriterium verwenden. Varianten ergeben sich dadurch, dass man $l > r$ oder $l < r$ wählt, die Parameter (l, r)

$m = 0,\ C^{(0)} = \emptyset,\ E^{(0)} = C$		
	/* **1.** Merkmalsmenge $C^{(m)}$ *vergrößern* /*	
	ermittle *bestes* noch nicht ausgewähltes Merkmal $c_{\nu(b)}$ relativ zu den vorhandenen: $m = m + 1,\ G(\{C^{(m-1)} \cup c_{\nu(b)}\}) = \max_{c_\nu \in E^{(m-1)}} G(\{C^{(m-1)} \cup c_\nu\})$	
	$C^{(m)} = C^{(m-1)} \cup c_{\nu(b)},\ E^{(m)} = E^{(m-1)} \backslash c_{\nu(b)}$	
	/* **2.** Merkmalsmenge $C^{(m)}$ bedingt *verkleinern* /*	
	Abbruchbedingung: $a = F$	
	WHILE $a = F \wedge \lvert C^{(m)}\rvert > 2$	
		ermittle *schlechtestes* bereits ausgewähltes Merkmal $c_{\nu(s)}$ relativ zu den vorhandenen: $G(\{C^{(m)} \backslash c_{\nu(s)}\}) = \max_{c_\nu \in C^{(m)}} G(\{C^{(m)} \backslash c_\nu\})$
		$i(*)$ sei maximales i mit $\lvert C^{(i)}\rvert = \lvert C^{(m)}\rvert - 1$
		IF $G(C^{(m)} \backslash c_{\nu(s)}) > G(C^{(i(*))})$
		THEN $m = m + 1,\ C^{(m)} = C^{(m-1)} \backslash c_{\nu(s)},\ E^{(m)} = E^{(m-1)} \cup c_{\nu(s)}$
		ELSE $a = T$
UNTIL $\lvert C^{(m)}\rvert = n$		
Ergebnis: $C^{(m)}$ mit n Merkmalen		

Bild 3.9.3: Eine zwischen Erweiterung und Reduktion der Merkmalsmenge alternierende Suche; begonnen wird mit einer leeren Menge von Merkmalen

konstant hält oder aber dynamisch variiert. Eine Möglichkeit ist, die Parameterwahl so einzuschränken, dass $|l - r| = 1$ ist. Experimentelle Ergebnisse aus der Literatur belegen, dass diese Suchverfahren sehr erfolgreich arbeiten, und zwar umso besser je größer die Parameter (l, r) gewählt werden; allerdings steigt damit der Suchaufwand.

7. Auswahl von n Merkmalen aus n' vorgegebenen mit der (l, r)–Suche für konstante Werte von (l, r). Wähle Werte für (l, r).

7.1 Wenn $l > r$ ist, initialisiere eine leere Menge von aktuell ausgewählten Merkmalen und beginne mit Schritt 7.2. Wenn $l < r$ ist, initialisiere als Menge von aktuell ausgewählten Merkmalen alle n' gegebenen Merkmale.

7.2 Wähle das Merkmal aus, das in der Menge der gegebenen Merkmale minus der schon ausgewählten Merkmale das beste ist. Wiederhole das l–mal. Beende die Suche, sobald die Menge von aktuell ausgewählten Merkmalen den Umfang n hat, sonst fahre fort mit Schritt 7.3.

7.3 Eliminiere das Merkmal, das am wenigstens zur Güte der Menge von aktuell ausgewählten Merkmalen beiträgt. Wiederhole das r–mal. Beende die Suche, sobald die Menge von aktuell ausgewählten Merkmalen den Umfang n hat, sonst fahre fort mit Schritt 7.2.

Abgesehen von der branch–and–bound Suche prüfen die obigen Suchverfahen nicht, ob die durch Hinzunahme eines Merkmals gebildete neue Menge *besser* ist als die alte Menge von Merkmalen. Vielmehr wird stets das Merkmal hinzugenommen, das unter den noch verbliebenen den größten Wert des Gütekriteriums liefert. Bei der **alternierenden Suche** ("floating search") wird die Zahl der zu eliminierenden Merkmale durch diesen Gesichtspunkt kontrolliert. Man braucht in dieser Variante der Suche *keine* Festlegung von Parametern (l, r) mehr. Es

wird der Suchalgorithmus skizziert, wenn man mit einer leeren Menge von ausgewählten Merkmalen beginnt und diese erweitert. Der andere Fall, dass man mit der Menge aller gegebenen Merkmale beginnt und diese reduziert, dürfte offensichtlich sein.

8. Auswahl von n Merkmalen aus n' vorgegebenen mit der alternierenden Suche.

8.1 Initialisiere eine leere Menge aktuell ausgewählter Merkmale.

8.2 Wähle das Merkmal aus, das relativ zur Menge der noch verbleibenden Merkmale am besten ist und füge es zur Menge aktuell ausgewählter Merkmale hinzu. Beende die Suche, wenn diese Menge n Merkmale enthält.

8.3 Ermittle das Merkmal, das zur Güte der Menge der aktuell ausgewählten Merkmale am wenigsten beiträgt.

Wenn die Güte der Menge der aktuell ausgewählten Merkmale durch Elimination dieses Merkmals *steigt*, eliminiere es und wiederhole Schritt 8.3. Sonst eliminiere es *nicht* und führe Schritt 8.2 aus.

In Bild 3.9.3 ist eine genauere Version einer alternierenden Suche gezeigt. Dabei ist $C = \{c_1, c_2, \ldots, c_{n'}\}$ die gegebene Menge mit n' Merkmalen, $C^{(m)}$ die Menge der ausgewählten Merkmale im Schritt m, $E^{(m)}$ die Menge der eliminierten Merkmale im Schritt m, $|C^{(m)}|$ die Zahl der Merkmale in $C^{(m)}$ und $G(C^{(m)})$ ein Mass für die Güte der Merkmale in $C^{(m)}$. Abgesehen vom Gütemaß G und der gewünschten Zahl n der Merkmale sind keine weiteren Parameter zu wählen. Bei diesem Algorithmus wird mit einer leeren Menge ausgewählter Merkmale begonnen und sukzessive weitere hinzugefügt. Der oben erwähnte Fall der sukzessiven Eliminierung von Merkmalen wird hier nicht genauer angegeben.

Merkmalsauswahl mit **evolutionärer Suche** basiert auf den in Abschnitt 1.6.10 kurz vorgestellten evolutionären Algorithmen, z. B. einer Evolutionsstrategie gemäß (1.6.38), S. 47. Eine Menge von Merkmalen bildet ein Element einer Population. Die Anfangspopulation P_μ^0 sind demnach μ anfänglich gewählte Mengen von Merkmalen. Aus diesen werden durch Kreuzung, Mutation und Selektion neue Populationen gebildet und nach einer Reihe von Schritten die beste, wiederum im Sinne eines der Gütekriterien aus Abschnitt 3.9.2, ausgewählt.

3.10 Symbole (VA.1.1.3, 13.04.2004)

3.10.1 Festlegung von Symbolen

Wie in Abschnitt 3.1 erläutert, geht es darum, ein Muster ${}^{\varrho}\boldsymbol{f}$ in eine **Symbolkette** ${}^{\varrho}v$ zu transformieren, wobei gemäß (3.1.6)

$$ {}^{\varrho}v = {}^{\varrho}v_1 {}^{\varrho}v_2 \ldots {}^{\varrho}v_{n({}^{\varrho})} , \quad {}^{\varrho}v_j \in V_T $$

ist, d. h. alle Elemente oder Symbole der Kette ${}^{\varrho}v$ sind aus einer vorgegebenen Menge V_T von einfacheren Bestandteilen des Musters genommen. Die Menge V_T ist so zu wählen, dass die Muster ${}^{\varrho}\boldsymbol{f} \in \Omega$ mit ausreichender Genauigkeit darstellbar sind.

Wie bei dem Merkmalsvektor ${}^{\varrho}\boldsymbol{c}$ kommt es auch bei der Symbolkette ${}^{\varrho}v$ weniger darauf an, dass die Muster möglichst vollständig dargestellt werden, als vielmehr darauf, dass die Klassen möglichst gut trennbar sind. Trotzdem wird praktisch ausschließlich die Darstellbarkeit der Muster mit Symbolen aus V_T als Kriterium verwendet. Bisher gibt es keine systematischen Ansätze zur Gewinnung und Bewertung von Symbolen, die denen in Abschnitt 3.8 oder in Abschnitt 3.9 vergleichbar wären. Daher ist dieser Abschnitt relativ kurz, obwohl der symbolischen Darstellung von Mustern in der Literatur allgemein eine große Bedeutung zugesprochen wird. Die Festlegung der Menge V_T, die auch als **terminales Alphabet**, Menge der **Grundsymbole** oder Menge der einfacheren Bestandteile bezeichnet wird, erfolgt heuristisch. Für ein– und zweidimensionale Muster gibt es dafür eine Reihe von Vorschlägen, von denen einige erläutert werden. Die Gemeinsamkeiten sind in den folgenden sieben Punkten zusammengefasst.

Bei relativ einfachen bildhaften Mustern geht man i. Allg. von zwei Vorstellungen aus:

1. Die für die Klassifikation wesentliche Information ist in der **Konturlinie** oder im Umriss des Objektes enthalten, sodass nur *linienhafte Muster* zu betrachten sind. Die Beispiele in Bild 3.10.1a zeigen, dass es verschiedenartige Muster gibt, für die diese Annahme zutrifft.
2. Man versucht, eine kleine Menge V_T von **Liniensegmenten** zu finden, aus denen die Linienmuster zusammengesetzt werden können. Wir bezeichnen diese Segmente auch als **Formelemente**. Bild 3.10.1b zeigt dafür einige Beispiele. Man zerlegt oder segmentiert also das Muster in seine Formelemente oder Grundsymbole.

Offensichtlich ist die Menge von Liniensegmenten *nicht* eindeutig. Man kann z. B. auf gekrümmte Liniensegmente ganz verzichten und Krümmungen mit kurzen, geraden Segmenten approximieren, wie es in Bild 2.1.11 geschieht. Neben der Darstellbarkeit des Musters sind weitere wichtige Gesichtspunkte für die Wahl des terminalen Alphabets:

3. Die einfacheren Bestandteile eines Musters müssen *extrahierbar* sein, d. h. es müssen Algorithmen bekannt sein, um in einem vorgegebenen Muster Elemente aus V_T zu finden.
4. Die nachfolgende Verarbeitung sollte *einfach* werden, insbesondere sollte sich ein einfacher Klassifikator bzw. ein einfacher Formalismus zur Beschreibung der Muster ergeben.

Für Beschreibungsformalismen und entsprechende Klassifikatoren wird auf die Literatur verwiesen, Klassifikatoren auf der Basis des Abstandes von Symbolketten werden in Abschnitt 4.6.4 behandelt.

Wenn die Annahmen 1. und 2. unzweckmäßig erscheinen, so ist eine nahe liegende Verallgemeinerung die folgende:

5. Statt einer Menge V_T von Liniensegmenten kann man auch Flächensegmente oder Volumensegmente als Formelemente wählen.

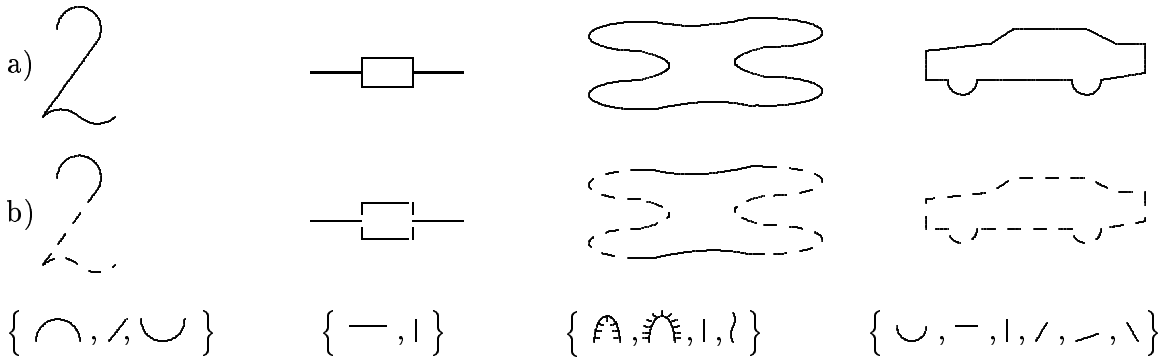


Bild 3.10.1: a) Einige Muster, deren Klassifikation aufgrund der Konturlinie möglich ist. b) Zerlegung in terminale Symbole (Grundsymbole) der in a) gezeigten Muster

Das ändert nichts an den Gesichtspunkten 3. und 4. Außer der Charakterisierung der Form von Liniensegmenten kann zur Darstellung eines Musters auch die Angabe der relativen Lage erforderlich sein:

6. Wenn zur eindeutigen Darstellung eines Musters die Aufeinanderfolge von Formelementen nicht hinreichend ist, so werden besondere Grundsymbole zur Kennzeichnung der gegenseitigen Lage von Formelementen eingeführt.

Wenn z. B. eine geschlossene Konturlinie, beginnend von einem bestimmten Startpunkt, mit Formelementen dargestellt wird, so reicht die Verkettung oder Aufeinanderfolge aus, und ein spezielles Symbol für diese eine **Lagerelation** ist unnötig. Dagegen gibt es beispielsweise in mathematischen Formeln wie (3.9.42) oder (3.9.44) außer der Aufeinanderfolge, also der Relation „neben“, auch noch andere Lagerelationen wie „über“ oder „rechts oben“. Dazu können i. Allg. weitere Relationen wie „enthalten in“, „umgeben von“, „benachbart zu“, „links von“, „unter“, „links unter“ usw. kommen.

7. Die Menge V_T der Grundsymbole oder das terminale Alphabet besteht aus der Menge der Formelemente vereinigt mit der Menge der Lagerelationen.

Bei eindimensionalen (wellenförmigen) Mustern sind zwei Fälle zu unterscheiden.

1. Das Muster ist relativ einfach, wie z. B. ein EKG. Dann wendet man die oben für linienhafte Muster beschriebene Vorgehensweise an. Bild 3.10.2a zeigt ein Beispiel.
2. Das Muster ist relativ komplex, wie z. B. Sprache. Wegen der großen Variabilität der Muster und der Datenfülle ist es meistens unzweckmäßig, den Funktionsverlauf mit terminalen Symbolen nachzubilden. Dann ordnet man größeren, durch je eine bestimmte Eigenschaft gekennzeichneten Bereichen des Musters ein Grundsymbol zu. Bild 3.10.2b zeigt auch dafür ein Beispiel. Diese Vorgehensweise ist im Prinzip natürlich auch auf Bilder übertragbar.

Bei zweidimensionalen (bildhaften) Mustern sind die meisten Grundsymbole dem Menschen auffällige Eigenschaften wie Linienende, Kreuzung, Krümmung, relatives Minimum, Wendepunkt, Unstetigkeit oder sonstige „kritische Punkte“ einer Linie. Einige der Literatur entnommene Beispiele sind in Bild 3.10.3 dargestellt. Man sieht, dass in der Regel nicht ausschließlich kurze gerade Liniensegmente unterschiedlicher Orientierung vorkommen, da diese

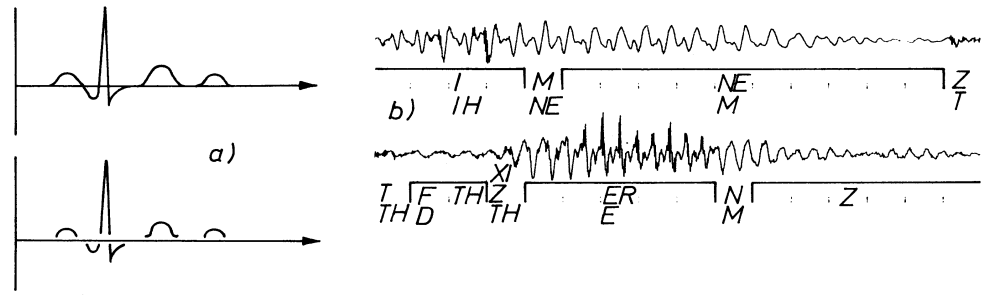


Bild 3.10.2: a) Ein eindimensionales Muster mit möglichen Grundsymbolen zur Darstellung des Funktionsverlaufs. b) Eine Darstellung der Worte „im Test“ und Ersetzung größerer Bereiche durch Grundsymbole

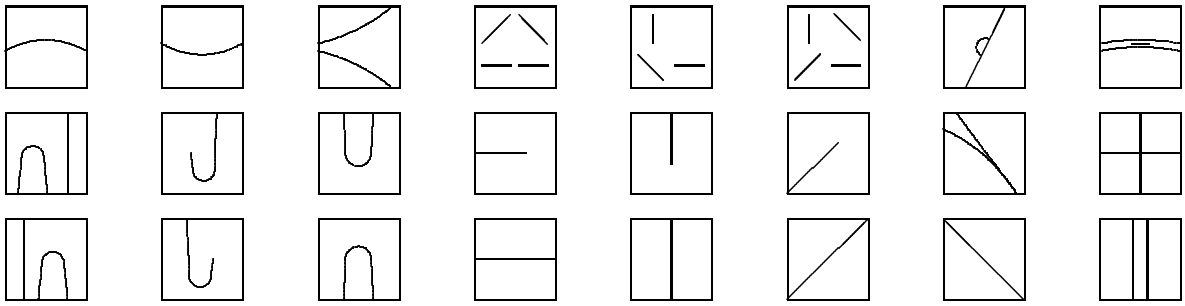


Bild 3.10.3: Beispiele für die Grundsymbole von linienhaften Mustern

zu wenig Struktur enthalten und die Beschreibung und Klassifikation dann zu kompliziert würde.

Wenn bei wellenförmigen Mustern die oben als Fall 1 erwähnte Methode der Darstellung des Funktionsverlaufs angewendet wird, so ergibt sich gegenüber den Formelementen von Bild 3.10.3 keine wesentliche Änderung. Um dagegen größeren Einheiten des Musters ein Grundsymbol zuzuordnen, sind Verfahren erforderlich, die eine parametrische Darstellung des Musters und möglicherweise auch eine numerische Klassifikation erfordern. Ein Beispiel ist die Berechnung einer Funktion aus dem Sprachsignal, die den zeitlichen Verlauf der Energie in einem bestimmten Frequenzband angibt. Nicht das Sprachsignal, sondern die Energiefunktion wird mit Grundsymbolen dargestellt, wobei in diesem Falle nur die vier Symbole „relatives Maximum bzw. Minimum“ und „ansteigender bzw. abfallender Funktionsteil“ verwendet werden. Man erreicht auf diese Weise zum einen eine Datenreduktion, da die Energiefunktion sich wesentlich langsamer ändert als das Sprachsignal, zum anderen den Übergang auf eine unmittelbar interpretierbare Funktion, da die Energie in einem geeigneten Frequenzbereich Hinweise auf Spracheigenschaften wie stimmhaft, stimmlos, Pause oder Plosivlaut gibt.

3.10.2 Extraktion von Symbolen

Aus dem letzten Abschnitt wurde deutlich, dass zu den grundlegenden Operationen bei der Extraktion von Grundsymbolen die Ermittlung gerader oder gekrümmter Linienelemente gehört. Dieses gilt sowohl für bildhafte als auch wellenförmige Muster, und es gilt sowohl für die Dar-

stellung des Musters selbst als auch einer aus dem Muster berechneten anderen Größe, wie z. B. der Energie eines Sprachsignals in einem Frequenzband.

Bei einem wellenförmigen Muster gibt die Folge der Abtastwerte $[f_j]$ direkt einen Kurvenverlauf an, mit dessen Approximation durch Formelemente unmittelbar begonnen werden kann. Dagegen muss ein bildhaftes Muster i. Allg. zunächst in ein linienhaftes transformiert werden. Da in diesem Band nur die Klassifikation relativ einfacher Muster behandelt wird, kann man meistens davon ausgehen, dass das Muster zunächst durch eine Schwellwertoperation – siehe Abschnitt 2.2 – vom Hintergrund trennbar ist. Man hat dann eine Folge $[f_{jk}]$, in der Objektpunkte den Wert 1 und Hintergrundpunkte den Wert 0 haben. Die Konturlinie des Musters ist durch eine Änderung der Funktionswerte von 0 auf 1 oder umgekehrt gekennzeichnet. Punkte auf der Konturlinie lassen sich mit folgendem einfachen Algorithmus ermitteln:

1. Gegeben ist eine binäre Folge $[f_{jk}]$, in der das Objekt durch den Funktionswert 1 charakterisiert ist. Die Matrix der Werte f_{jk} wird zeilenweise durchsucht, bis der erste Punkt P mit dem Wert 1 gefunden wird. Dieses ist der erste Punkt auf der Konturlinie.
2. Man stelle sich vor, dass man auf den zuletzt erreichten Punkt zugegangen ist. Wenn er den Wert 1 hat, biege man nach links ab, sonst nach rechts. Jeder Punkt mit dem Wert 1, der dabei erreicht wird, liegt auf der Kontur.
3. Man wiederhole Schritt 2, bis das Objekt umfahren ist.
4. Das Ergebnis ist eine geordnete Liste von Konturpunkten, die dadurch entsteht, dass man das Objekt, beginnend bei P, so umläuft, dass das Objektinnere zur Rechten liegt.

Eine andere Vorgehensweise besteht darin, zunächst Konturpunkte zu finden und sie dann zu ordnen. Jeder Punkt f_{jk} mit dem Wert 1, der weniger als acht Nachbarpunkte mit dem Wert 1 hat, ist ein Konturpunkt.

Ausgangspunkt der weiteren Verarbeitung ist eine *geordnete Menge*

$$S = \{(x_j, y_j) \mid j = 1, 2, \ldots, N\} \tag{3.10.1}$$

von Wertepaaren. Bei einem eindimensionalen Muster $f(x)$ ist $x_j = j\Delta x$ der quantisierte Wert der unabhängigen Variablen, $y_j = f(x_j)$ der zugehörige Funktionswert. Die Ordnung ergibt sich nach ansteigenden Werten von x_j. Für ein zweidimensionales Muster sind x_j, y_j die Koordinaten eines auf der Konturlinie liegenden Punktes, und die Ordnung ergibt sich aus der Reihenfolge der Punkte bei einem Umlauf um die Konturlinie. Ein Beispiel zeigt Bild 3.10.4.

Vielfach wird eine gemäß (3.10.1) gegebene Kurve zunächst *stückweise linear* approximiert. Dafür ist es erforderlich, geeignete Geraden durch vorgegebene Punkte zu legen. Die Gleichung einer Geraden durch die beiden Punkte (x_j, y_j) und (x_k, y_k) ist

$$\begin{aligned} x(y_j - y_k) + y(x_k - x_j) &= y_j(x_k - x_j) - x_j(y_k - y_j)\ , \\ ax + by &= c\ , \qquad a^2 + b^2 > 0\ . \end{aligned} \tag{3.10.2}$$

Der **Ordinatenabstand** d_j eines Punktes (x_j, y_j) von der obigen Geraden ist definiert durch

$$d_j = \frac{ax_j + by_j - c}{b} \tag{3.10.3}$$

und der **senkrechte Abstand** s_j ist definiert durch

$$s_j = \frac{|ax_j + by_j - c|}{\sqrt{a^2 + b^2}}\ . \tag{3.10.4}$$

Für die stückweise Approximation muss die Punktmenge S in Teilmengen zerlegt werden

$$\begin{aligned} S &= \{S_1, \ldots, S_i, \ldots, S_m\} \ , \\ S_i &= \{(x_j, y_j) \mid j = 1, 2, \ldots, N_i\} \ . \end{aligned} \tag{3.10.5}$$

Als *Fehler* der Approximation einer Punktmenge S_ν kann der *maximale Abstand*

$$\varepsilon_{a,i}(d) = \max_{\{j \mid (x_j,y_j)\in S_i\}} |d_j| \ , \tag{3.10.6}$$

$$\varepsilon_{a,i}(s) = \max_{\{j \mid (x_j,y_j)\in S_i\}} s_j \tag{3.10.7}$$

oder der *mittlere quadratische Abstand*

$$\varepsilon_{m,i}(d) = \frac{1}{N_i} \sum_{\{j \mid (x_j,y_j)\in S_i\}} |d_j| \ , \tag{3.10.8}$$

$$\varepsilon_{m,i}(s) = \frac{1}{N_i} \sum_{\{j \mid (x_j,y_j)\in S_i\}} s_j \tag{3.10.9}$$

gewählt werden. Natürlich kann bei der Approximation einer endlichen Menge von Punkten durch Geraden der Fehler stets zu Null gemacht werden, indem man je zwei Punkte durch eine Gerade approximiert. Um viele kurze Geradenstücke zu vermeiden, darf also der zulässige Fehler nicht unrealistisch klein vorgegeben werden. Ein Fehlermaß, das sowohl den Approximationsfehler als auch die Länge der Geraden g_j und ihre Zahl m berücksichtigt, ist z. B.

$$\varepsilon_{\mathrm{k}} = \sum_{i=1}^{m} \frac{\varepsilon_{m,i}(s) m}{N_i} \ , \tag{3.10.10}$$

das bezüglich m, N_i, $\varepsilon_{m,i}(s)$ zu minimieren ist. Die hier am Beispiel der Approximation durch Geraden gemachten Ausführungen gelten analog für andere parametrische Kurven. Für eine Kurve mit der allgemeinen expliziten Gleichung

$$y = g(x, \boldsymbol{a}) \ , \tag{3.10.11}$$

bei der die Funktion g bis auf den Parametervektor $\boldsymbol{a}$ bekannt ist, wird in der Regel aus Gründen der einfachen Berechenbarkeit der Parameter das Fehlermaß

$$\varepsilon_{\mathrm{e}} = \frac{1}{N_i} \sum_{\{j \mid (x_j,y_j)\in S_i\}} (y_j - g(x_j, \boldsymbol{a}))^2 \tag{3.10.12}$$

verwendet.

Ein einfaches Verfahren, um eine Punktmenge S gemäß (3.10.1), die zu einem eindimensionalen Funktionsverlauf gehört, stückweise durch Geraden zu approximieren, ist das folgende:

1. Die Punktmenge S soll stückweise linear approximiert werden, wobei der Fehler $\varepsilon_{a,i}(d)$ oder $\varepsilon_{a,i}(s)$ in keinem Stück einen Schwellwert θ überschreitet.
2. Anfangs– und Endpunkt der ersten Geraden sind (x_1, y_1) und (x_N, y_N).
3. Wenn für alle Geraden der Fehler nicht größer ist als θ, dann ist die Approximation *beendet*, sonst fahre fort mit Schritt 4.
4. Für jede der bisher gefundenen Geraden mit zu großem Fehler führe Schritt 5 und 6 aus.

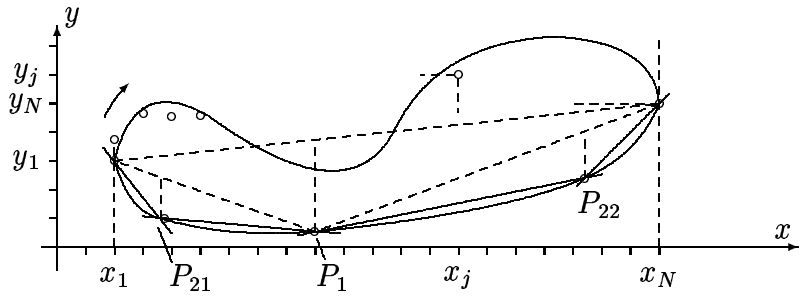


Bild 3.10.4: Darstellung der Kontur eines Objektes durch eine geordnete Punktmenge und stückweise lineare Approximation derselben. Wegen der Quantisierung liegen die Punkte i. Allg. nicht genau auf der Konturlinie

5. Ermittle den Punkt P mit größtem Abstand von einer der Geraden.
6. Ersetze diese alte Gerade durch zwei neue Geraden. Anfangs– und Endpunkt der ersten neuen Geraden sind der Anfangspunkt der alten Geraden und P, für die zweite neue Gerade sind dieses P und der Endpunkt der alten Geraden.
7. Gehe zurück nach Schritt 3.

Da für die (geschlossene) Konturlinie eines zweidimensionalen Musters $(x_1, y_1) = (x_N, y_N)$ ist, lässt sich dieses Verfahren nicht ohne weiteres darauf anwenden. Jedoch ist dieses in einfachen Fällen durch folgende Modifikation möglich. Man wählt als Punkte (x_1, y_1) und (x_N, y_N) die Berührungspunkte der Kurve mit den am weitesten auseinander liegenden Seiten des umschreibenden Rechtecks. Der obere und untere Zweig der Kurve werden nun mit dem obigen Algorithmus getrennt approximiert. In Bild 3.10.4 werden also sowohl der obere als auch der untere Zweig anfänglich von der Geraden durch (x_1, y_1) und (x_N, y_N) approximiert. Für den unteren Zweig ist P_1 der Punkt mit dem größten Ordinatenabstand d. Im nächsten Schritt wird der untere Zweig mit je einer Geraden durch (x_1, y_1) und P_1 sowie durch P_1 und (x_N, y_N) approximiert. Dieses Verfahren wird solange fortgesetzt, bis die verlangte Genauigkeit erreicht wird. Es ist möglich, dass einige Geradenstücke sehr kurz werden, jedoch lässt sich das, falls gewünscht, durch zusätzliche Bedingungen vermeiden.

Die in Bild 3.10.4 zwischen (x_1, y_1) und P_1 liegende Menge S_i von Punkten auf dem unteren Kurvenzweig wurde hier einfach mit einer Geraden durch (x_1, y_1) und P_1 approximiert. Eine bessere Approximation erhält man natürlich, wenn man eine **Ausgleichsgerade** für S_i berechnet. Der Rechenaufwand wird allerdings größer. Da bei einer geschlossenen Konturlinie sowohl fast horizontale $(a \simeq 0)$ als auch fast vertikale $(b \simeq 0)$ Geradenstücke möglich sind, empfiehlt sich die Anwendung des senkrechten Abstandes $\varepsilon_a(s)$ oder $\varepsilon_m(s)$ als Fehlermaß oder die Einführung einer Fallunterscheidung. Im ersteren Falle wird statt (3.10.2) die HESSE-Normalform der Geradengleichung verwendet, im zweiten Falle wird statt d_j in (3.10.3) der nicht normierte Abstand

$$d_j \cdot b = ax_j + by_j - c \tag{3.10.13}$$

als Fehlermaß definiert. Zur Abkürzung setzen wir

$$\overline{u} = \sum_{j=1}^{N_i} u_j \ , \quad \overline{u}^2 = \left(\sum_{j=1}^{N_i} u_j \right)^2 \ , \quad \overline{u^2} = \sum_{j=1}^{N_i} u_j^2 \tag{3.10.14}$$

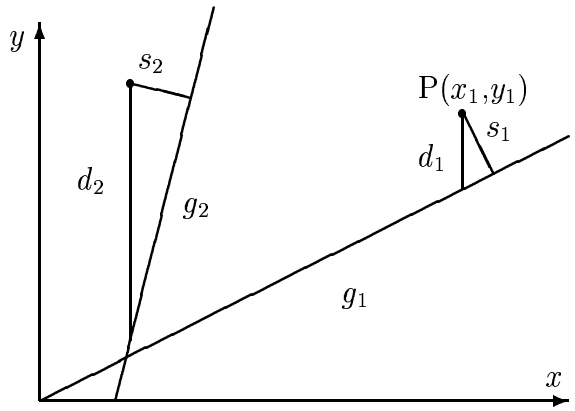


Bild 3.10.5: Ordinatenabstand d und senkrechter Abstand s bei Geraden unterschiedlicher Steigung

und berechnen für die Punkte $(x_j, y_j) \in S_i$ die Größe

$$p = \overline{y}^2 - \overline{x}^2 - N_i(\overline{y^2} - \overline{x^2}) \; . \tag{3.10.15}$$

Die Koeffizienten in (3.10.2) sind für $p \geq 0$

$$\begin{aligned} a &= N_i\, \overline{xy} - \overline{x}\,\overline{y} \; , \\ b &= \overline{x}^2 - N_i\, \overline{x^2} \; , \\ c &= \frac{1}{N_i}(a\overline{x} + b\overline{y}) \end{aligned} \tag{3.10.16}$$

und für $p \leq 0$

$$\begin{aligned} a &= \overline{y}^2 - N_i\, \overline{y^2} \; , \\ b &= N_i\, \overline{xy} - \overline{x}\,\overline{y} \; , \\ c &= \frac{1}{N_i}(a\overline{x} + b\overline{y}) \; . \end{aligned} \tag{3.10.17}$$

Bei nahezu horizontalen oder vertikalen Geraden ergeben die beiden Ansätze merkliche Unterschiede in den Geradengleichungen. Das liegt daran, dass der nicht normierte Ordinatenabstand (3.10.13), der in (3.10.16) verwendet wird, bei fast vertikalen Geraden ein ungeeignetes Maß ist, wie auch aus Bild 3.10.5 hervorgeht, während in (3.10.17) der entsprechend definierte, nicht normierte Abszissenabstand verwendet wird.

Ein anderer Algorithmus zur stückweise linearen Approximation einer Kurve, der sich insbesondere bei komplizierteren Konturlinien empfiehlt, ist der **Zerlege-und-vereinige-Algorithmus** ("split-and-merge" Algorithmus). Ausgangspunkt ist wieder die geordnete Menge S in (3.10.1).

1. Man wähle eine anfängliche Zerlegung $S_i^0, \;\; i = 1, \ldots, m_0$ von S in Teilmengen, eine parametrische Familie von Funktionen zur Approximation jeder Teilmenge S_i^0 – z. B. die Familie der Geraden in (3.10.2) – sowie ein Fehlermaß ε zur Bewertung der Approximationsgüte – z. B. den Fehler (3.10.6) – und einen zulässigen Schwellwert θ für den Fehler.

2. Im ν–ten Schritt wird die Zerlegung von S mit $S_i^\nu,\ i = 1, \ldots, m_\nu$ bezeichnet, der Fehler bei der Approximation der Punkte aus S_i^ν mit ε_i^ν.
3. Suche ein S_i^ν, dessen Fehler $\varepsilon_i^\nu \geq \theta$ ist, *zerlege* dieses S_i^ν in $S_j^{\nu+1}, S_{j+1}^{\nu+1}$ und approximiere $S_j^{\nu+1}$ und $S_{j+1}^{\nu+1}$ (damit wird gegenüber der Anfangszerlegung die Zahl der zu approximierenden Teilmengen erhöht). Wiederhole diesen Schritt, bis für alle Teilmengen der Fehler kleiner als θ wird.
4. Suche ein Paar S_i^ν, S_{i+1}^ν, nach dessen *Vereinigung* zu einer neuen Teilmenge der Fehler der Approximation kleiner als θ bleibt. Wiederhole diesen Schritt, bis keine weiteren Vereinigungen von Teilmengen mehr möglich sind.
5. Reduziere den Fehler bei fester Anzahl m_ν vom Teilmengen.
   5.1. Für alle Paare $S_i^\nu, S_{i+1}^\nu,\ i = 1, \ldots, m_\nu - 1$ berechne den Fehler $s = \max\{\varepsilon_i^\nu, \varepsilon_{i+1}^\nu\}$ und führe Schritt 5.2 – 5.4 aus.
   5.2. Mache versuchsweise den letzten Punkt aus S_i^ν zum ersten Punkt aus S_{i+1}^ν. Berechne für die so modifizierten Teilmengen den Fehler s'.
   5.3. Mache versuchsweise den ersten Punkt aus S_{i+1}^ν zum letzten Punkt aus S_i^ν. Berechne für die so modifizierten Teilmengen den Fehler s''.
   5.4. Wähle die Teilmengen, die zu $\min\{s, s', s''\}$ führen.
   5.5. Wiederhole obige Operation, bis keine Teilmengen mehr verändert werden.

Beginnend mit einer beliebigen Anfangszerlegung der zu approximierenden Punktmenge werden also zunächst alle *Zerlegungen* ausgeführt, die erforderlich sind, den vorgegebenen Fehler einzuhalten, dann alle *Vereinigungen*, die ohne Überschreitung der Fehlergrenze möglich sind, und schließlich werden die *Grenzen* zwischen Teilmengen so verschoben, dass der Fehler vermindert wird. Durch die Wahl unterschiedlicher parametrischer Familien von approximierenden Funktionen und anderer Fehlermaße sind zahlreiche Varianten möglich. Es sei noch angemerkt, dass der Zerlege-und-vereinige-Algorithmus auch bei der Analyse komplexer Muster zur Darstellung von Konturen, Ermittlung von Regionen und Erfassung von Textureigenschaften angewendet wird.

Das Ergebnis der stückweise linearen Approximation eines Musters $f(x)$ oder der Kontur eines Musters $f(x, y)$ ist eine Folge von Liniensegmenten oder Vektoren $V_i,\ i = 1, \ldots, m$. Jedes Segment ist gekennzeichnet durch seinen Startpunkt und die Parameter a_i, b_i, c_i der Geradengleichung (3.10.2), oder durch den Startpunkt P_i, den Winkel α_i gegenüber der x–Achse und die Länge l_i. Diese Segmente können verwendet werden, um daraus Formelemente der in Bild 3.10.1 gezeigten Art aufzubauen, z. B. quadratische Kurve (eine Folge etwa gleich langer, miteinander etwa gleich große Winkel einschließender Liniensegmente), Linie (ein oder mehrere fast kollineare Segmente), Unterbrechung (ein oder zwei sehr kurze Segmente), Ecke (zwei Linien unter bestimmtem Winkel mit oder ohne Unterbrechung) oder Linienzug (zwei Linien mit kleinem eingeschlossenen Winkel).

Zur Charakterisierung einer Kurve wird vielfach die lokale Krümmung K verwendet.

Definition 3.17 *Ist α der Winkel der Tangente an die Kurve mit der x–Achse und l die Bogenlänge, so ist die* **Krümmung** *definiert als*

$$K = \frac{\mathrm{d}\alpha}{\mathrm{d}l} = \frac{\dfrac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2}}{\sqrt{\left(1 + \dfrac{\mathrm{d}f(x)}{\mathrm{d}x}^2\right)^3}} \tag{3.10.18}$$

und der **Krümmungsradius** *ist die reziproke Krümmung.*

In einer quantisierten Kurve wird die Krümmung näherungsweise dadurch bestimmt, dass man die Differenziale durch Differenzen ersetzt. Sind z. B. zwei etwa gleich lange benachbarte Liniensegmente mit den Winkeln α_1 und α_2 gegenüber der x–Achse gegeben, so ist die Krümmung im Schnittpunkt der Segmente

$$K \simeq \alpha_2 - \alpha_1 \ , \tag{3.10.19}$$

wobei $\Delta l = 1$ Längeneinheit gesetzt wurde. Die Liniensegmente, mit denen die Winkeldifferenz und damit die Krümmung berechnet wird, sollten einige Bildpunkte lang sein, um den Einfluss von Störungen zu reduzieren. Die Krümmung wird genutzt, um an Ecken, d. h. an Stellen besonders großer Krümmung, bei der stückweisen Approximation mit Geraden oder gekrümmten Kurvenzügen zwangsweise ein Linienende einzufügen.

3.11 Literaturhinweise

Entwicklung nach einer Orthogonalbasis

Eine Einführung in Orthogonalbasen und Frames gibt [Pei und Yeh, 1997]. Umfangreichere Texte sind [Albert, 1972, Gantmacher, 1958]. Die Frames wurden in [Duffin und Schaeffer, 1952] eingeführt. Die Berechnung des Frameoperators wird in [Kaiser, 1994] behandelt.

Eine Erweiterung der orthogonalen Entwicklungen sind die in [Bovik et al., 1992, Pattichis et al., 2001] behandelten amplituden- und frequenzmodulierten Entwicklungen, die vor allem bei nichtstationären Mustern Vorteile bieten. Da die Verwendung von Datenfenstern für Zwecke der Musterklassifikation oft hinreichend ist, wurden diese Entwicklungen hier nicht behandelt. Die in [Agaian et al., 2001] eingeführten sequenzgeordneten Transformationen wurden bereits in Abschnitt 2.7 erwähnt.

Die FOURIER-Transformation ist einer der immer wieder genutzten Ansätze, wie z. B. aus [Persoon und Fu, 1977, Persoon und Fu, 1986, Gray und Goodman, 1995, Lai et al., 2001] hervorgeht; die verallgemeinerte Version wird z. B. in [Barshan und Ayrulu, 2002] genutzt. Zum Cesptrum wird auf [Oppenheim und Schafer, 1975, Braun, 1975], zu homomorphen Systemen auf [Niemann, 1981, Oppenheim, 1968, Oppenheim und Schafer, 1975] verwiesen. Die Berechnung des Leistungsspektrums wird in Chap. 11 von [Oppenheim und Schafer, 1975] diskutiert. Zur schnellen FOURIER-Transformation gibt es eine Vielzahl von Arbeiten, z. B. [Ahmed und Rao, 1975, Brigham, 1995, Brigham, 1997, Granlund, 1972, Nussbaumer, 1981] sowie die bereits in Abschnitt 2.7 erwähnten; neben dem hier beschriebenen Ansatz gibt es die rekursiven Verfahren [Goertzel, 1958, Chan et al., 1994, Yang und Chen, 2002]. Die Transformation der Konturlinie wird in [Granlund, 1972, Zahn und Roskies, 1972] durchgeführt. Zu skalen- und rotationsinvarianten Merkmalen auf der Basis der MELLIN-Transformation wird auf [Casasent und Psaltis, 1977, Götze et al., 1999, Jin et al., 2004] verwiesen. Affininvariante FOURIER-Merkmale werden in [Arbter et al., 1990, Kuthirummal et al., 2004] vorgestellt. Einen experimentellen Vergleich macht [Kauppinen et al., 1995]. Die diskrete cosinus-Transformation (DCT) wird in [Ahmed et al., 1974, Ahmed und Rao, 1975, Docef et al., 2002] behandelt. Eine Erweiterung sind Merkmale aus dem Bispektrum [Chandran et al., 1997].

Die WALSH-Funktionen werden in [Ahmed und Rao, 1975, Harmuth, 1970] beschrieben, die HWT in [Ahmed und Rao, 1975]. Für weitere Transformationen wie Slant-HADAMARD oder HARTLEY-Transformation wird auf [Agaian und Duvalyan, 1991, Agaian et al., 2004, Bracewell, 1983, Bracewell, 1985, Ghurumuruhan und Prabhu, 2004] verwiesen.

Wavelet-Transformation

Die Basis der *Wavelet-Transformation* wurde in [Haar, 1910] gelegt. Eine grundlegende Arbeit zu dem Thema ist [Grossmann und Morlet, 1984]. Umfassende Darstellungen enthalten die Bücher [Burrus et al., 1998, Chui, 1992, Daubechies, 1992, Kaiser, 1994], zusammenfassende Darstellungen [Chan und Liu, 1998, Rioul und Vetterli, 1991]. Die Erweiterung der Skalierungsfunktion zu einem Vektor von Funktionen wird in [Kleinert, 2004] ausführlich beschrieben. Die Bestimmung der Waveletkoeffizienten in der feinsten Auflösung wird in [Burrus et al., 1998] diskutiert, sowie kurz in [Chan und Liu, 1998]; dort wird auch die Projektion von Abtastwerten in den approximierenden Unterraum bei nichtorthogonalen Wavelets erörtert. Ergebnisse zur Faltung für die kontinuierliche Wavelettransformation werden in [Pérez-Rendón und Robles, 2004] vorgestellt.

Die Ergebnisse in Zusammenhang mit (3.3.10) bis (3.3.17), S. 188, sind z. B. in [Mallat, 1989] als Theoreme formuliert, deren Beweise skizziert werden. Die in Abschnitt 3.3.3 erwähnte Abtastung mit Wavelets wird in [Chan und Liu, 1998] analysiert. Das Problem der Transformation periodischer diskreter Funktionen wird in [Burges, 1998, Gubner und Chang, 1995] behandelt.

Bezüge zwischen Skalierungsfunktion und B-Splines werden in [Unser und Blu, 2003b] hergestellt, die Transformation mit beliebigem (nicht notwendig ganzzahligem) Skalierungsindex in [Muñoz et al., 2002]. Eine sphärische Transformation wird in [Pastor et al., 2001] für die Darstellung dreidimensionaler Objekte genutzt. Eine komplexe Wavelet-Transformation wird in [Kingsbury, 1998, Magarey und Kingsbury, 1998] eingeführt, ganzzahlige Varianten in [Adams und Ward, 2003, Li et al., 2005]. Invariante Merkmale auf der Basis der Wavelet-Transformation werden in [Khalil und Bayoumi, 2001, Khalil und Bayoumi, 2002] entwickelt (s. auch die Angaben bei Texturerkennung). Die Berechnung optimaler Wavelets erfolgt in [Bosnyakov und Obukhov, 2002, Coifman und Wickerhauser, 1992, Saito, 1994, Saito und Coifman, 1995]. Zu Verallgemeinerungen wie biorthogonalen Wavelets und "Wavelet Packets" wird auf [Cohen et al., 1992, Vetterli und Herley, 1992, Meyer, 1993] verwiesen, zu M-Band Wavelets und cosinus-modulierten auf [Gopinath und Burrus, 1992, Gopinath und Burrus, 1995]. Tabellen von Wavelets und Filtern sind in [Daubechies, 1988, Daubechies, 1993, Johnston, 1980, Smith und Barnwell, 1986] angegeben.

Die Eigenschaften der in JPEG 2000 verwendeten Wavelets werden in [Unser und Blu, 2003a] untersucht; Übersichten findet man in [Skodras et al., 2001, Usevitch, 2001]. In [Figueiredo und Nowak, 2001] wird Bildvorverarbeitung waveletbasiert durchgeführt. Die Kantenberechnung in Bildern mit Hilfe der Wavelet-Transformation erfolgt in [Grossman, 1986, Mallat und Zhong, 1992, Mallat und Hwang, 1992]. Die Nutzung von Wavelets für die Funktionsapproximation mit neuronalen Netzen wird in [Zhang und Benveniste, 1992] gezeigt. Waveletbasierte Merkmale für die Sprechererkennung werden in [Lung, 2004a, Lung, 2004b] kurz untersucht, für die Gesichtserkennung in [Chien und Wu, 2002].

Filterbänke

Die GABOR-*Filter* gehen auf [Gabor, 1946] zurück und wurden in [Daugman, 1985] auf Bilder übertragen. Die oben eingeführte komplexe Version wird z. B. in [Bovik et al., 1990, Dunn und Higgins, 1995] verwendet, der Realteil in [Jain und Farrokhnia, 1991]. Der Beweis zu Satz 3.5, S. 198, ist z. B. in [Dunn und Higgins, 1995] angegeben. Ansätze zur automatisierten Wahl der Filterparameter findet man in [Dunn und Higgins, 1995, Teuner et al., 1995, Bresch, 2002]. Die Beziehungen (3.4.8), (3.4.9), (3.4.10) und (3.4.11) sind [Bovik et al., 1990] entnommen. Eine Kombination von GABOR-Filter, Hauptachsentransformation und Analyse unabhängiger Komponenten wird in [Liu und Wechsler, 2003] für die Gesichtserkennung vorgestellt, invariante GABOR-Merkmale in [Kyrki et al., 2004].

Zu GAUSS-Filtern wird auf [Burt und Adelson, 1983, Schiele und Pentland, 1999, Schmid und Mohr, 1996, Sporring, 1997] verwiesen.

Andere Ansätze, die hier nicht behandelt wurden, Spline-Filter [Unser et al., 1991, Unser et al., 1993a, Unser et al., 1993b, Panda und Chatterji, 1997] und steuerbare Filter [Anderson, 1992, Freeman und Adelson, 1991, Huang und Chen, 1995, Jacob und Unser, 2004, Simoncelli und Freeman, 1995, Simoncelli und Farid, 1996, Yu, 2001].

Andere heuristische Verfahren

Die R-Transformation wurde in [Reitboeck und Brody, 1969] eingeführt.

Momenteninvarianten wurden in [Hu, 1962] eingeführt und u. a. in [Alt, 1962, Dudani et al., 1977, Wong und Hall, 1978, Reiss, 1991] vorgestellt; weitere Arbeiten dazu sind [Abu-Mostafa und Psaltis, 1984, Abu-Mostafa und Psaltis, 1985, Balslev et al., 2000, Candocia, 2004, Flusser und Suk, 1993, Flusser et al., 2003, Kan und Srinath, 2002, Prokop und Reeves, 1992, Reeves et al., 1988, Rothe et al., 1996, Sadjadi und Hall, 1980, Reiss, 1993, Suk und Flusser, 2003, Teague, 1980, Teh und Chin, 1988]. In [van Gool et al., 1996, Mindru et al., 1999] sind Momente angegeben, die invariant gegen eine affine Transformation und photometrische Verzerrungen (Helligkeit, Kontrast) sind, in [Suk und Flusser, 2003] solche, die invariant gegen eine affine Transformation und eine lineare Verzerrung mit einer punktsymmetrischen Impulsantwort sind. Ein experimenteller Vergleich verschiedener Momenteninvarianten wurde in [Belkasim et al., 1991] vorgenommen. Zur effizienten Berechnung von Momenten wird auf [Belkasim und Kamel, 2001, Flusser, 2000, Jiang und Bunke, 1991, Li, 1991, Mukundan und Ramakrishnan, 1995, Tuzikov et al., 2003, Yang und Albregtsen, 1994] verwiesen.

Es gibt zahlreiche Varianten der Definition von Momenten wie z. B. gewichtete [Balslev et al., 2000], ZERNICKE- [Khotanzad, 1990, Mukundan und Ramakrishnan, 1995], LEGENDRE- [Mukundan und Ramakrishnan, 1995, Chong et al., 2004], TCHEBICHEV- [Mukandan et al., 2001] oder KRAWTCHOUK- [Yap et al., 2003] Momente; die letzten beiden erlauben eine diskrete Formulierung. Momente (dreidimensionaler) Polyhedra werden in [Li, 1993, Liggett, 1988, Saupe und Vranić, 2001, Sheynin und Tuzikov, 2001, Sloane und Conway, 1982] berechnet bzw. genutzt. Die Approximation von polygonalen Konturen mit Momenten erfolgt in [Shu et al., 2002].

Angepasste Filter für den kontinuierlichen Fall werden in Abschnitt 5.5 von [Winkler, 1977] oder in Sect. 16.3 von [Middleton, 1960] behandelt, der diskrete Fall in [Arcese et al., 1970] oder in Chap. 19 von [Pratt, 1991]. Phasenangepasste Filter werden in [Chen et al., 1994] behandelt. Erweiterungen werden in [Brunelli und Poggio, 1997] vorgestellt, Anwendungen zur Detektion von Gesichtern in [Govindaraju, 1996], die Verwendung von Schablonen für Objekt und Nicht-Objekt in [Shaick und Yaroslavski, 2001, Sung und Poggio, 1998].

Eckenmerkmale werden in [Büker und Hartmann, 1996] untersucht.

Zu den hier nicht behandelten *fraktalen Merkmalen* wird auf [Chen et al., 1993, Keller et al., 1989, Peleg et al., 1984, Pentland, 1984, Sarkar und Chaudhuri, 1992, Tan und Yan, 2001] verwiesen.

Merkmale für die Texturerkennung

Übersichten zur Texturerkennung geben [van Gool et al., 1985, Haralick, 1979, Randen und Husøy, 1999, Wagner, 1999, Zhang und Tan, 2002]. Einen Standardsatz von Texturbildern enthält [Brodatz, 1966]. Menschliche Texturerkennung wird in [Caelli und Julesz, 1978, Julesz, 1975, Julesz, 1982, Julesz und Bergen, 1983, Julesz, 1986] untersucht.

Ein Vergleich von 18 Typen von Texturmerkmalen an 7 Typen von Texturen wird mit Angabe von Erkennungsrate und Rechenzeit in [Wagner, 1999] gegeben; ein Vergleich von etwa 20 Typen von Texturmerkmalen an 3 Typen von Texturen erfolgt in [Randen und Husøy, 1999].

Merkmale, die wenigstens an einem der in [Wagner, 1999] untersuchten Typen von Texturen die besten Erkennungsraten erzielten, sind danach die in [Galloway, 1975, Unser, 1986b, Chen et al., 1995, Amelung, 1995] angegebenen, wobei zudem die Rechenzeit bei Merkmalen nach [Galloway, 1975, Unser, 1986b, Amelung, 1995] mit zu den geringsten gehört. Merkmale, die an einem Typ von Texturen unter den drei besten waren, sind zusätzlich [Laine und Fan, 1993, Laws, 1980, Pikaz und Averbuch, 1997]. Die höchsten Rechenzeiten ergeben sich bei Merkmalen nach [Haralick et al., 1973, Laws, 1980, Chen et al., 1995]. Weitere Vergleiche enthalten [Fountain et al., 1998, Pichler et al., 1996, Rubner et al., 2001].

Texturmerkmale auf der Basis der GABOR-Funktionen wurden vorgeschlagen und untersucht z. B. in [Dunn et al., 1994, Fogel und Sagi, 1989, Grigorescu et al., 2002, Jain und Farrokhnia, 1991, Manthalkar et al., 2003b, Teuner et al., 1995, Pichler et al., 1996, Panda und Chatterji, 1997]. Ein Vorschlag zur automatischen Bestimmung der Filterparameter wird in [Dunn und Higgins, 1995, Teuner et al., 1995] gemacht, erfordert allerdings zusätzliche Rechenzeit. Weitere Arbeiten dazu sind [Aach et al., 1995, Bovik et al., 1990, Clausi und Jernigan, 2000]. Histogramme der Ausgabe von Filterbänken werden in [Liu und Wang, 2003] untersucht.

Texturmerkmale auf der Basis der Wavelet-Transformation, darunter auch rotationsinvariante, werden in [Charlampidis und Kasparis, 2002, Chen et al., 1994, Do und Vetterli, 2002, Kim und Udpa, 2000, Manthalkar et al., 2003a, Pun und Lee, 2003, Pun, 2003] vorgeschlagen.

Weitere Arbeiten sind die Verwendung der fraktalen Dimension [Chaudhury und Sarkar, 1995], von autoregressiven Modellen [Kashyap und Khotanzed, 1986], von FOURIER- und HADAMARD-Merkmalen [Azencott et al., 1997, Haley und Manjunath, 1999, Unser, 1986a, Unser und Eden, 1989], von bildpunktbasierten Merkmalen [Park et al., 2004], der HAAR-Transformation [Lonnestad, 1992], von morphologischen Merkmalen [Li et al., 1996], Assoziationsregeln [Rushing et al., 2001], Lauflängen [Chu et al., 1990] oder neuronalen Netzen für Merkmalsgewinnung und Klassifikation [Jain und Karu, 1996]. Ein Maß für den Texturunterschied wird in [Li und Leung, 2002] vorgeschlagen. Auch die Nutzung des Entropieprinzips (s. Abschnitt 4.2.4) und von MARKOV-Ketten wurde vorgeschlagen [Zhu et al., 1997, Zhu et al., 1998, Zhu et al., 2000]. In [Kim et al., 2002b] wird die Verwendung der Grauwerte als Eingabe für eine Support Vektor Maschine experimentell untersucht.

Texturmerkmale für das Wiederfinden von Bildern werden in [Çarkacioğlu und Yarman-Vural, 2003] vorgeschlagen.

Merkmale für die Spracherkennung

Modelle der Sprachproduktion werden in [Berry et al., 1995, Fant, 1960, Flanagan, 1972, Flanagan, 1983, Story und Titze, 1998] entwickelt. Zur linearen Vorhersage in der Sprachverarbeitung wird auf [Itakura und Saito, 1970, Makhoul, 1975, Markel und Gray Jr., 1982, White und Neeley, 1976] verwiesen, für andere Signale und Bilder auf [Bohlin, 1973, Deguchi und Morishita, 1978, Tjostheim und Sandvin, 1979, Wood und Treitel, 1975]. Die hier verwendete Autokorrelationsmethode wird in [Chandra und Lin, 1974] mit der Kovarianzmethode verglichen und hat i. Allg. Vorteile hinsichtlich Stabilität und Effizienz bei vergleichbarer Analysegenauigkeit. Die LEVINSON-Rekursion [Levinson, 1947] wird in Sect. 3.3 von [Markel und Gray Jr., 1976] gezeigt, wo auch genauer auf die theoretische Begründung des Modellspektrums eingegangen wird. Zur Ermittlung der Zahl

m der Vorhersagekoeffizienten wird weiter auf [Parzen, 1974] verwiesen. Die Präemphase in (3.6.22) stammt aus [Wakita, 1973]. Modifikationen der linearen Vorhersage sind in [Hermansky et al., 1985, Hernando und Nadeu, 1991, Hernando und Nadeu, 1997, Hernando et al., 1997, Mansour und Juang, 1989, McGinn und Johnson, 1983] zu finden. Zu Ansätzen für eine nichtlineare Modellierung auf der Basis der Strömungstheorie wird auf [Teager, 1980, Thomas, 1986] verwiesen.

Physiologische Grundlagen der Sprachwahrnehmung werden in [Keidel und Neff, 1974, 1975, 1976, Plomp, 1964, Zwicker und Feldtkeller, 1967] gelegt, dort wird u. a. auch die mel-Frequenzskala begründet. Weitere Literatur dazu ist [Allen und Neely, 1997, Buser und Imbert, 1992, Picone, 1993, Yost, 1994]. Verschiedene Filter zur Bildung der Frequenzgruppen werden in [Hermansky et al., 1986, Ruske, 1979, Regel, 1988, Seneff, 1986, Rieck, 1995] untersucht. Ein Modell der Basilarmembran wird in [Strube, 1985] angegeben. Die mel-Cepstrum Koeffizienten wurden in [Davis und Mermelstein, 1980] eingeführt; weitere Arbeiten dazu sind [Chow et al., 1987, Niemann et al., 1988, Pols, 1977, Regel, 1988, Rieck, 1995, Seidel, 1974], wobei in [Pols, 1977, Seidel, 1974] eine Hauptachsentransformation verwendet wurde, die in [Davis und Mermelstein, 1980] durch die nun praktisch ausschließlich verwendete diskrete cosinus-Transformation ersetzt wurde. Eine Optimierung der Filterparameter für die mel-Cepstrum Koeffizienten erfolgt in [Lee et al., 2003]. Als Merkmale für eine gegen Störgeräusche robuste Worterkennung hat das root-Cepstrum, das in [Lim, 1979] eingeführt wurde, Vorteile; es wurde in [Alexandre und Lockwood, 1993, Karnjanadecha und Zahorian, 2001, Sarikaya, 2001, Sarikaya und Hansen, 2001, Wu et al., 1991, Yapanel et al., 2001] weiter untersucht. Die Vorteile einer getrennten Berechnung von mel-Cepstrum Koeffizienten in verschiedenen Frequenzbändern werden in [Hariharan et al., 2001] aufgezeigt. Vergleichende Untersuchungen werden in [Jankowski et al., 1995, Regel, 1988, Rieck, 1995, Schless, 1999] durchgeführt. Weitere Einzelheiten zu den Normierungsverfahren in Abschnitt 3.6.5 sind in [Bu und Chiueh, 2000, Class et al., 1993, Lerner und Mazor, 1992, Schless, 1999, Viikki und Laurila, 1997, Zhao et al., 1995] enthalten.

In [Zhou et al., 2001] wurde ein nichtlinearer Ansatz für die Klassifikation unter Stress vorgestellt, der auf dem TEAGER-Energieoperator [Kaiser, 1990, Kaiser, 1993, Teager, 1980] beruht. Halbsilben werden in [Plannerer und Ruske, 1992, Ruske und Schotola, 1978, Ruske und Schotola, 1981, Ruske, 1994, Schotola, 1984] und silbenbasierte Merkmale in [Chen et al., 2002] entwickelt und untersucht. Ein Merkmalssatz, der das auditive System genauer modelliert, ist in [Li et al., 2000, Mak et al., 2004] beschrieben.

Merkmale für die Objekterkennung

Histogramme als Basis für die Erkennung von Objekten werden in [Mel, 1997, Schwartz und Austin, 1991, Schiele und Crowley, 1996, Schiele und Crowley, 2000] genutzt. Die in Abschnitt 3.7.2 vorgestellten lokalen Merkmale wurden ausführlich in [Pösl, 1999, Reinhold, 2003] untersucht. Merkmale auf der Basis wichtiger Punkte wurden in [Lowe, 2004] vorgestellt.

Die kombinierten Merkmale in Abschnitt 3.7.3 wurden in [Mel, 1997] beschrieben.

Analytische Methoden

Die zum Beweis von Satz 3.8 genutzte Extremaleigenschschaft der Eigenvektoren ist in [Gantmacher, 1958] (S. 291–299) zu finden; für den kontinuierlichen Fall wird auf [Niemann, 1974] (S. 101–109) verwiesen. Die KARHUNEN-LOÈVE-Entwicklung geht auf [Karhunen, 1946, Loève, 1955] zurück, ihre Nutzung in der Merkmalsgewinnung auf [Watanabe, 1965]. Die Minimierung des Approximationsfehlers durch sie und die Dekorrelation der Koeffizienten wird in [Middleton, 1960], Sect. 8.2-2, oder [Watanabe, 1965] gezeigt. Die Ausführungen in Abschnitt 3.8.2 beruhen im Wesentlichen auf [Niemann, 1969, Niemann, 1970, Niemann, 1974, Niemann und Winkler, 1969]. Die speziellen Faktorisierungen (3.8.16) – (3.8.21) findet man in [Hornegger et al., 2000, Murase und Lindenbaum, 1995]. Zahlreiche Anwendungen, Erweiterungen und Modifikationen gehen aus [Baumberg und Hogg, 1994, Borotschnig et al., 2000, Capelli et al., 2001, Cootes et al., 1993, Hummel, 1979, Hornegger et al., 2000, Kim et al., 2002a, Kittler, 1975, Kuhn et al., 2000, Leonardis und Bischof, 1996, Ozeki, 1979, Murase und Nayar, 1995, Sirovich und Kirby, 1987, Therrien, 1975, Turk und Pentland, 1991, Yilmaz und Gökmen, 2001] hervor. Experimentelle Ergebnisse zum Kriterium s_4, $s_{5,1}$ sind in [Belhumeur et al., 1997, Niemann, 1971, Hornegger et al., 2000] zu finden, Untersuchungen zur (näherungsweisen) Normalverteilung in [Niemann, 1970]. Die Diskriminanzanalyse wird z. B. in [Fukunaga, 1990] genauer erörtert, Varianten in [Xu et al., 2004, Ye und Li, 2004]. Der Bezug zu den gemeinsamen Vektoren (“common vectors”) wird in [Gülmezoğlu et al., 2001] dargestellt. Ein Vergleich der Hauptachsentransformation mit der Diskriminanzanalyse wird in [Martinez und Kak, 2001] durchgeführt und zeigt, dass erstere durchaus Vorteile haben kann. Zweidimensionale Erweiterungen der Hauptachsentransformation und der linearen Diskriminanzanalyse werden in [Li und Yuan, 2005, Yang et al., 2004, Yang et al., 2005] vorgeschlagen. Die verallgemeinerte Hauptachsentransformation zur Bestimmung stückweise linearer Approximationen wird in [Vidal et al., 2005] entwickelt.

Zur numerischen Berechnung von Eigenwerten und -vektoren symmetrischer Matrizen wird auf [Wilkins, 1962/63, Press et al., 1994, Press et al., 2002] verwiesen, für neuronale und sequentielle Algorithmen, die insbesondere nicht die Berechnung der Kovarianzmatrix erfordern, auf [Oja, 1982, Oja, 1983, Oja, 1992, Sanger, 1989, Wang et al., 2003, Weng et al., 2003]; Die Hauptachsentransformation ist ein Beispiel für die *Einbettung* von (hochdimensionalen) Daten in einen niedriger dimensionalen Raum; eine Übersicht über Einbettungsverfahren gibt [Hjaltason und Samet, 2003].

Die nichtlineare (kernbasierte) Hauptachsentransformation wird z. B. in [Schölkopf et al., 1999, Müller et al., 2001, Park et al., 2004, Yang, 2002] behandelt, die kernbasierte Diskriminanzanalyse in [Mika et al., 1999, Baudat und Anouar, 2000, Yang, 2002]. Die dort verwendete Berechnung mit der Matrix $\boldsymbol{T}$ der Skalarprodukte transformierter Vektoren wird auch in [Kirby und Sirovich, 1990] aufgezeigt. Sie findet zunehmend Verwendung, z. B. in der Objektverfolgung [Comaniciu et al., 2003] oder Merkmalsgewinnung [Girolami, 2002, Mika et al., 2003, Titsias und Likas, 2001]. Nichtlineare Hauptachsentransformation mit neuronalen Netzen wird in [Kramer, 1991, Malthouse, 1998] durchgeführt.

Die als „klassifikatorbezogene Merkmalsauswahl“ bezeichnete Vorgehensweise von Abschnitt 3.8.4 wird in [de Figueiredo, 1976, Jäpel, 1980] untersucht. Bedingungen für die Existenz optimaler Parameter in (3.8.77) werden in [de Figueiredo, 1974] abgeleitet. Die Gleichungen (3.8.80) bzw. (3.8.84), (3.8.87) sind [Wilks, 1962] bzw. [van Otterloo und Young, 1978], [Decell und Quirein, 1973] entnommen. Die Methode der projizierten Gradienten wird in

[Horst, 1979] beschrieben. Die Ergebnisse in Bild 3.8.9 sind aus [Jäpel, 1980, Reinhardt, 1979], die Vorverarbeitung der isoliert gesprochenen Wörter wird in [Becker, 1974] beschrieben.

Eine Darstellung von Verfahren der Analyse unabhängiger Komponenten enthalten [Hyvärinen et al., 2001, Oja et al., 2004]; sie wird mit analytischen Verfahren z. B. in [Cao et al., 2003, Grbic et al., 2001, Molgedy und Schuster, 1994, Plumbley, 2003, Plumbley und Oja, 2004] behandelt, mit neuronalen Ansätzen z. B. in [Bell, 1995, Kasprzak und Cichocki, 1996], mit der charakteristischen Funktion in [Eriksson und Koivunen, 2003]. In [Boscolo et al., 2004] wird eine PARZEN-Schätzung der Verteilungsdichte zur Identifikation der Komponenten verwendet. Weitere Arbeiten zur Analyse unabhängiger Komponenten sind [Common, 1994, Hyvärinen und Oja, 1997, Hyvärinen, 1999, Hyvärinen, 2001, Pham, 2001, Regalia und Kofidis, 2003, Wu und Chiu, 2001]. In [Bartlett, 2001, Bartlett et al., 2002] wird sie zur Gesichtserkennung eingesetzt, in [Le Borgne et al., 2004] allgemein zur Merkmalsgewinnung. Der unterbestimmte Fall wird in [Bofill und Zibulevsky, 2001] behandelt. Die Kombination von Analyse unabhängiger Komponenten mit Mischungsverteilungen (s. Abschnitt 4.8.2) erfolgt in [Lee und Lewicki, 2002], die Kombination mit der Singulärwertzerlegung in [Vrabie et al., 2004]..

Ein hier nicht behandelter Ansatz ist die in [Lee und Seung, 1999] eingeführte nichtnegative Matrixfaktorisierung [Guillamet et al., 2003].

Merkmalsbewertung und -auswahl

Das Auswahlproblem wird in [Ben-Bassat, 1980, Cover, 1974, Cover und Van Campenhout, 1977] untersucht. Die Abschätzungen (3.9.4) und (3.9.7) und Vermutungen dazu sind aus [Devijver, 1974, Chen, 1975], die Abschätzung (3.9.12) aus [Devijver, 1974, Niemann, 1969]. Die Berechnung des BAYES-Abstandes mit der PARZEN-Schätzung wird in [Fu et al., 1970] angegeben. Zu den Gütemaßen (3.9.14) – (3.9.16) wird auf [Bhattacharyya, 1943, Kailath, 1967, Vajda, 1970, Lewis, 1962, Vilmansen, 1973] verwiesen, zu denen in (3.9.17) – (3.9.22) auf [Vajda, 1970, Lissack und Fu, 1976, Chernoff, 1952, Matusita, 1966, Patrick und Fisher, 1969], sowie für weitere auf [Ben-Bassat, 1978, Kullback, 1968, Mucciardi und Gose, 1971, Vilmansen, 1973]. Die Abschätzungen (3.9.29) – (3.9.31) sind aus [Chen, 1975, Fukunaga, 1990, Lissack und Fu, 1976], zu (3.9.33) s. [Fukunaga, 1990]. Die Aussage im Zusammenhang mit (3.9.41) wird in [Heydorn, 1971] gezeigt. Eine Weiterentwicklung der Transinformation gibt [Kwak und Choi, 2002b]. Die Berechnung von Schätzwerten der Transinformation mit der PARZEN-Schätzung wird in [Kwak und Choi, 2002a] durchgeführt.

Übersichten über Algorithmen zur Merkmalsauswahl geben [Dash und Liu, 1997, Kittler, 1978, Siedlecki und Sklansky, 1988], vergleichende Untersuchungen [Jain und Zongker, 1997, Kudo und Sklansky, 2000]. Die heuristischen Auswahlverfahren sind [Bakis et al., 1968, Mucciardi und Gose, 1971, Whitney, 1971] entnommen. Zu dynamischer Programmierung wird auf Abschnitt 1.6.8 sowie [Bellman und Kalaba, 1965, Chang, 1973, Cheung und Eisenstein, 1978, Schneeweiß, 1974] verwiesen, zur branch-and-bound Suche auf [Chen, 2003, Iannarilli und Rubin, 2003, Korbut und Finkelstein, 1971, Lawler und Wood, 1966, Narendra und Fukunaga, 1977]; einige weitere Ansätze liefern [Garnett und Yau, 1977, Ichino, 1981, Jain und Dubes, 1978, Marril und Green, 1963, Patrick und Fisher, 1969, Stearns, 1976, Whitney, 1971]. Die hier als gleitende Suche bezeichnete Methode wurde in [Pudil et al., 1994] angegeben (dort als SFFS-Algorithmus, “sequential forward floating selection”, bezeichnet) und als sehr guter

Kompromiss zwischen Rechenzeit und Merkmalsgüte bezeichnet.

Merkmalsauswahl durch Messung der Merkmalsähnlichkeit wird in [Mitra et al., 2002] betrachtet.

Invarianten

Eine klassische Arbeit ist die über Momenteninvarianten [Hu, 1962], der weitere zu diesem Thema folgten, in denen Momente mit unterschiedlichen Invarianzeigenschaften abgeleitet werden, [Abu-Mostafa und Psaltis, 1984, Belkasim et al., 1991, Dudani et al., 1977, Flusser und Suk, 1993, Flusser et al., 2003, Kan und Srinath, 2002, Khotanzad, 1990, Li, 1991, Mukundan und Ramakrishnan, 1995, Prokop und Reeves, 1992, Sadjadi und Hall, 1980, Teague, 1980, Wong und Hall, 1978, Yang und Albregtsen, 1994, Zhu et al., 2002]. Dazu kommen Arbeiten zur Rotationsinvarianz [Choi und Kim, 2002, Tsai und Tsai, 2002], zu differenziellen Invarianten [Rothwell et al., 1992], kombinierten Invarianten [Rothwell et al., 1992], affin invarianten FOURIER- [Abter et al., 1990, Oirrak et al., 2002, Oirrak et al., 2003], Polynom- [Keren et al., 1994] und Waveletkoeffizienten [Khalil und Bayoumi, 2001, Khalil und Bayoumi, 2002] sowie weitere affin invariante Merkmale [Matsakis et al., 2004, Petrou und Kadyrov, 2004], spektralen Invarianten [Chandran et al., 1997, Götze et al., 1999, Lai et al., 2001] und anderen [Besl und Jain, 1986, Burkhardt, 1979, Burel und Hénocq, 1995, Keysers et al., 2000, Keysers et al., 2004, Mundy und Zisserman, 1992, Suk und Flusser, 1996, Schulz-Mirbach, 1994, Schulz-Mirbach, 1995, Weiss, 1993, Zhang und Tan, 2002].

Symbole

Die Beispiele für Grundsymbole von bildhaften Mustern lehnen sich an [Blum und Nagel, 1978, Freeman, 1978, Ledley, 1964, Ledley, 1972, Moayer und Fu, 1975, Moayer und Fu, 1976, Mori et al., 1970, Sheinberg, 1970, Shelman, 1972, Stallings, 1972] an, die für wellenförmige an [Rivoira und Torasso, 1978, Stockman, 1978]. Der Konturverfolgungsalgorithmus in Abschnitt 3.10.2 geht auf [Niemann, 1974] zurück, die Fallunterscheidung mit (3.10.15) wird in [Niemann, 1974] (Abschnitt 2.2.2) gezeigt. Ein verbesserter Algorithmus zur Konturverfolgung wird in [Ren et al., 2002] vorgestellt. Die Algorithmen zur stückweise linearen Approximation wurden in [Bley, 1982, Niemann, 1981, Pavlidis und Horowitz, 1974, Ramer, 1972] entwickelt. Die Zusammenfassung von Geradenstücken zu komplexeren Formelementen wird in [Pavlidis und Ali, 1979] durchgeführt. Die Berechnung von Krümmungen bei diskreten Mustern wird in [Rosenfeld und Johnston, 1973, Rosenfeld, 1975, Bennett und Mac Donald, 1975] behandelt. Eine experimentelle Untersuchung verschiedener Krümmungsdefinitionen enthält [Harbeck, 1996].

Literaturverzeichnis

[Aach et al., 1995] Aach, T., Kaup, A., Mester, R. On texture analysis: Local energy transforms versus quadrature filters. *Signal Processing*, 45:173–181, 1995.

[Abter et al., 1990] Abter, K., Snyder, W.E., Burkhardt, H., Hirzinger, G. Application of affine-invariant Fourier descriptors to recognition of 3-D objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:640–647, 1990.

[Abu-Mostafa und Psaltis, 1984] Abu-Mostafa, Y.S., Psaltis, D. Recognitive aspects of moment invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:698–706, 1984.

[Abu-Mostafa und Psaltis, 1985] Abu-Mostafa, Y.S., Psaltis, D. Image normalization by complex moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7:46–55, 1985.

[Adams und Ward, 2003] Adams, D., Ward, R.K. Symmetric-extension-compatible reversibel integer-to-integer wavelet transforms. *IEEE Trans. on Signal Processing*, 51:2624–2636, 2003.

[Agaian und Duvalyan, 1991] Agaian, S., Duvalyan, V. On slant transforms. *Pattern Recognition and Image Analysis*, 1:317–326, 1991.

[Agaian et al., 2004] Agaian, S., Tourshan, K., Noonan, J.P. Generalized parametric Slant-Hadamard transform. *Signal Processing*, 84(8):1299–1306, 2004.

[Agaian et al., 2001] Agaian, S.S., Panetta, K., Grigoryan, A.M. Transform-based image enhancement algorithms with performance measure. *IEEE Trans. on Information Theory*, 47:367–382, 2001.

[Ahmed et al., 1974] Ahmed, N., Natarajan, T., Rao, K.R. Discrete cosine transform. *IEEE Trans. on Computers*, 23:88–93, 1974.

[Ahmed und Rao, 1975] Ahmed, N., Rao, K. *Orthogonal Transforms for Digital Signal Processing*. Springer, Berlin, Heidelberg, New York, 1975.

[Albert, 1972] Albert, A. *Regression and the Moore-Penrose Pseudoinverse*, S. 15–23. Academic, New York, 1972.

[Alexandre und Lockwood, 1993] Alexandre, P., Lockwood, P. Root cepstral analysis: A unified view, applications to speech processing in car environments. *Speech Communication*, 12:277–288, 1993.

[Allen und Neely, 1997] Allen, J., Neely, S. Modelling the relation between the intensity JND and loudness for pure tones and wide-band noise. *Journal of the Acoustical Society of America*, 102(6):3628–3646, 1997.

[Alt, 1962] Alt, F.L. Digital pattern recognition by moments. In G.L. Fischer, D.K. Pollock, B. Raddack, M.E. Stevens, Hg., *Optical Character Recognition*, S. 153–179. Spartan Books, Washington, 1962.

[Amelung, 1995] Amelung, J. *Automatische Bildverarbeitung für die Qualitätssicherung*. Dissertation, Technische Hochschule Darmstadt, Darmstadt, Germany, 1995. Darmstädter Dissertationen D17.

[Anderson, 1992] Anderson, M.T. *Controllable Multidimensional Filters and Models in Low Level Computer Vision*, Bd. 282 von *Linköping Studies in Science and Technology. Dissertations*. Department of Electrical Engineering, Linköping University, Linköping, Schweden, 1992.

[Arbter et al., 1990] Arbter, K., Snyder, W., Burkhardt, H., Hirzinger, G. Application of affine-invariant Fourier descriptors to recognition of 3D objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:640–647, 1990.

[Arcese et al., 1970] Arcese, A., Mengert, P.H., Trombini, E.W. Image detection through bipolar correlation. *IEEE Trans. on Information Theory*, 16:534–541, 1970.

[Azencott et al., 1997] Azencott, R., Wang, J., Younes, L. Texture classification using windowed Fourier filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19:148–153, 1997.

[Bakis et al., 1968] Bakis, R., Herbst, N.M., Nagy, G. An experimental study of machine recognition of handprinted numerals. *IEEE Trans. on Systems Science and Cybernetics*, 4:119–132, 1968.

[Balslev et al., 2000] Balslev, I., Doring, K., Eriksen, R.D. Weighted central moments in pattern recognition. *Pattern Recognition Letters*, 21:381–384, 2000.

[Barshan und Ayrulu, 2002] Barshan, B., Ayrulu, B. Fractional Fourier transform pre-processing for neural networks and its application to object recognition. *Neural Networks*, 15(1):131–140, 2002.

[Bartlett, 2001] Bartlett, M.S. *Face Image Analysis by Unsupervised Learning*. Kluwer Academic Publishers, Boston, USA, 2001.

[Bartlett et al., 2002] Bartlett, M.S., Movellan, J.R., Sejnowski, T.J. Face recognition by independent component analysis. *IEEE Trans. on Neural Networks*, 13:1450–1464, 2002.

[Baudat und Anouar, 2000] Baudat, G., Anouar, F. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):40–42, 2000.

[Baumberg und Hogg, 1994] Baumberg, A., Hogg, D. Learning flexible models from image sequences. In J.O. Eklundh, Hg., *Proc. European Conference on Computer Vision (ECCV)*, S. 316–327. Springer LNCS 801, 1994.

[Becker, 1974] Becker, D. Vergleich eines linearen und eines nichtlinearen Klassifikators bei der Worterkennung. *Wissensch. Berichte AEG-Telefunken*, 47:77–84, 1974.

[Belhumeur et al., 1997] Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19:711–720, 1997.

[Belkasim und Kamel, 2001] Belkasim, S., Kamel, M. Fast computation of 2-D image moments using biaxial transform. *Pattern Recognition*, 34:1867–1877, 2001.

[Belkasim et al., 1991] Belkasim, S.O., Shridhar, M., Ahmadi, M. Pattern recognition with moment invariants: A comparative study and new results. *Pattern Recognition*, 24:1117–1138, 1991.

[Bell, 1995] Bell, T.E. Harvesting remote sensing data. *IEEE Spectrum*, 32(3):24–31, 1995.

[Bellman und Kalaba, 1965] Bellman, R., Kalaba, R. *Dynamic Programming and Modern Control Theory*. Academic Press, New York, 1965.

[Ben-Bassat, 1978] Ben-Bassat, M. f-entropies, probability of error, and feature selection. *Information and Control*, 39:227–242, 1978.

[Ben-Bassat, 1980] Ben-Bassat, M. On the sensitivity of the probability of error rule for feature selection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:57–60, 1980.

[Bennett und Mac Donald, 1975] Bennett, J.R., Mac Donald, J.S. On the measurement of curvature in a quantized environment. *IEEE Trans. on Computers*, 24:803–820, 1975.

[Berry et al., 1995] Berry, D.A., Herzel, H., Titze, I.R., Krischer, K. Interpretation of biomechanical simulations of normal and chaotic vocal fold oscillations with empirical eigenfunctions. *Journal of the Acoustical Society of America*, 96:3595–3604, 1995.

[Besl und Jain, 1986] Besl, P.J., Jain, R.C. Invariant surface characteristics for 3-d object recognition in range images. *Computer Vision, Graphics, and Image Processing*, 33:33–80, 1986.

[Bhattacharyya, 1943] Bhattacharyya, A. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Mathematical Society*, 35(3):99–110, 1943.

[Bley, 1982] Bley, H. *Vorverarbeitung und Segmentierung von Stromlaufplänen unter Verwendung von Bildgraphen*. Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 1982.

[Blum und Nagel, 1978] Blum, H., Nagel, R.N. Shape discrimination using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

[Bofill und Zibulevsky, 2001] Bofill, P., Zibulevsky, M. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81:2353–2362, 2001.

[Bohlin, 1973] Bohlin, T. Comparison of two methods of modeling stationary EEG signals. *IBM J. Res. Dev.*, S. 194–205, 1973.

[Borotschnig et al., 2000] Borotschnig, H., Paletta, L., Prantl, M., Pinz, A. Appearance-based active object recognition. *Image and Vision Computing*, 18:715–727, 2000.

[Boscolo et al., 2004] Boscolo, R., Pan, H., Roychowdhury, V.P. Independent component analysis based on a nonparametric density estimation. *IEEE Trans. on Neural Networks*, 15:55–65, 2004.

[Bosnyakov und Obukhov, 2002] Bosnyakov, M.S., Obukhov, Y.V. Optimal wavelet basis for functions satisfying heat equation. In *Proc. 6. Int. Conference on Pattern Analysis and Image Processing: New Information Technologies (PRIA-6-2002*, S. 108–110. Russian Academy of Science, Velikiy Novgorod, Russian Federation, 2002.

[Bovik et al., 1990] Bovik, A.C., Clark, M., Geisler, W.S. Multichannel texture analysis using localized spatial filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:55–73, 1990.

[Bovik et al., 1992] Bovik, A.C., Gopal, N., Emmoth, T., Restrepo, A. Localized measurement of emergent image frequencies by Gabor wavelets. *IEEE Trans. on Information Theory*, 38:691–712, 1992.

[Bracewell, 1983] Bracewell, R.N. Discrete Hartley transform. *J. Optical Society of America*, 73:1832–1835, 1983.

[Bracewell, 1985] Bracewell, R.N. *The Hartley Transform*. Oxford University Press, Oxford, England, 1985.

[Braun, 1975] Braun, S. Signal analysis for rotating machinery vibrations. *Pattern Recognition*, 7:81–86, 1975.

[Bresch, 2002] Bresch, M. Optimizing filter banks for supervised texture recognition. *Pattern Recognition*, 35:783–790, 2002.

[Brigham, 1995] Brigham, E.O. *FFT Schnelle Fourier-Transformation*. R. Oldenbourg Verlag, München, Germany, 6. Aufl., 1995.

[Brigham, 1997] Brigham, E.O. *FFT Anwendungen*. R. Oldenbourg Verlag, München, Germany, 6. Aufl., 1997.

[Brodatz, 1966] Brodatz, P. *Textures*. Dover, New York, 1966.

[Brunelli und Poggio, 1997] Brunelli, R., Poggio, T. Template matching: Matched spatial filters and beyond. *Pattern Recognition*, 30:751–768, 1997.

[Bu und Chiueh, 2000] Bu, L., Chiueh, T.-D. Perceptual speech processing and phonetic feature mapping for robust vowel recognition. *IEEE Trans. on Speech and Audio Processing*, 8:105–114, 2000.

[Büker und Hartmann, 1996] Büker, U., Hartmann, G. Eckenmerkmale für die robuste Erkennung und Fovealisierung in einem Robot Vision System. In B. Jähne, P. Geißler, H. Haußecker, F. Hering, Hg., *Mustererkennung 1996*, S. 590–597. Springer (Berlin, Heidelberg), 18. DAGM Symposium, Heidelberg, 1996.

[Burel und Hénocq, 1995] Burel, G., Hénocq, H. Three-dimensional invariants and their application to object recognition. *Signal Processing*, 45:1–22, 1995.

[Burges, 1998] Burges, C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[Burkhardt, 1979] Burkhardt, H. *Transformationen zur lageinvarianten Merkmalgewinnung*. Fortschrittberichte, Reihe 10. VDI Verlag, Düsseldorf, Germany, 1979.

[Burrus et al., 1998] Burrus, C.S., Gopinath, R.A., Guo, H. *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1998.

[Burt und Adelson, 1983] Burt, P.J., Adelson, E.H. The Laplacian pyramid as a compact image code. *IEEE Trans. Communication*, 31:532–540, 1983.

[Buser und Imbert, 1992] Buser, P., Imbert, M. *Audition*. MIT Press, Cambridge, MA, USA, 1992.

[Caelli und Julesz, 1978] Caelli, T., Julesz, B. On perceptual analyzers underlying visual texture discrimination. *Biological Cybernetics*, 29:201–214, 1978.

[Candocia, 2004] Candocia, F.M. Moment relations and blur invariant conditions for finite-extent signals in one, two, and n-dimensions. *Pattern Recognition Letters*, 25:437–447, 2004.

[Cao et al., 2003] Cao, J., Murata, N., Amari, J.-I., Cichocki, A., Takeda, T. A robust approach to independent component analysis of signals with high-level noise measurements. *IEEE Trans. on Neural Networks*, 14:631–645, 2003.

[Capelli et al., 2001] Capelli, R., Maio, D., Maltoni, D. Multispace KL for pattern representation and classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:977–996, 2001.

[Çarkacioğlu und Yarman-Vural, 2003] Çarkacioğlu, A., Yarman-Vural, F. SASI: A generic texture descriptor for image retrieval. *Pattern Recognition*, 36:2615–2633, 2003.

[Casasent und Psaltis, 1977] Casasent, D., Psaltis, D. New optical transforms for pattern recognition. *Proc. IEEE*, 65:77–84, 1977.

[Chan und Liu, 1998] Chan, A.K., Liu, S.J. *Wavelet Toolware: Software for Wavelet Training*. Academic Press, San Diego, USA, 1998.

[Chan et al., 1994] Chan, Y.H., Chau, L.P., Siu, W.C. Efficient implementation of discrete cosine transform using recursive filter structure. *IEEE Trans. Circuit Syst. Video Tech.*, 4:550–552, 1994.

[Chandra und Lin, 1974] Chandra, S., Lin, W.C. Experimental comparison between stationary and nonstationary formulations of linear prediction applied to voiced speech signals. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 22:403–415, 1974.

[Chandran et al., 1997] Chandran, V., Carswell, B., Boashash, B., Elgar, S. Pattern recognition using invariants defined from higher order spectra: 2D images. *IEEE Trans. on Image Processing*, 6:703–712, 1997.

[Chang, 1973] Chang, C.Y. Dynamic programming as applied to feature subset selection in a pattern recognition system. *IEEE Trans. on Systems, Man, and Cybernetics*, 3:167–171, 1973.

[Charlampidis und Kasparis, 2002] Charlampidis, D., Kasparis, T. Wavelet-based rotational invariant roughness features for texture classification and segmentation. *IEEE Trans. on Image Processing*, 11:825–837, 2002.

[Chaudhury und Sarkar, 1995] Chaudhury, B.B., Sarkar, N. Texture segmentation using fractal dimensions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:72–77, 1995.

[Chen et al., 2002] Chen, B., Wang, H., Lee, L. Discriminating capabilities of syllable-based features and approaches of utilizing them for voice retrieval of speech information in Mandarin Chinese. *IEEE Trans. on Speech and Audio Processing*, 10:303–314, 2002.

[Chen, 1975] Chen, C.H. On a class of computationally efficient feature selection criteria. *Pattern Recognition*, 7:87–94, 1975.

[Chen et al., 1994] Chen, Q., Defrise, M., Deconinck, F. Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1156–1168, 1994.

[Chen et al., 1993] Chen, S.S., Keller, J.M., Crownover, R.M. On the calculation of fractal features from images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:1087–1090, 1993.

[Chen, 2003] Chen, X. An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24:1925–1933, 2003.

[Chen et al., 1995] Chen, Y., Nixon, M., Thomas, D. Statistical geometrical features for texture classification. *Pattern Recognition*, 28:537–552, 1995.

[Chernoff, 1952] Chernoff, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[Cheung und Eisenstein, 1978] Cheung, R.S., Eisenstein, B.A. Feature selection via dynamic programming for text-independent speaker identification. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 26:397–403, 1978.

[Chien und Wu, 2002] Chien, J.-T., Wu, C.C. Discriminant waveletfaces and nearest feature classifiers

for face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1644–1649, 2002.

[Choi und Kim, 2002] Choi, M.-S., Kim, W.-Y. A novel two stage template matching method for rotation and illumination invariance. *Pattern Recognition*, 35:119–129, 2002.

[Chong et al., 2004] Chong, C.-W., Raveendran, P., Mukundan, R. Translation and scale invariants of Legendre moments. *Pattern Recognition*, 37:119–129, 2004.

[Chow et al., 1987] Chow, Y., Dunham, M., Kimball, O., Krasner, M., Kubala, G.F., Makhoul, J., Price, P., Roucos, S., Schwartz, R. BYBLOS: The BBN continuous speech recognition system. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 89–92. Dallas, Texas, 1987.

[Chu et al., 1990] Chu, A., Sehgal, C.M., Greenleaf, J.F. Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters*, 11:415–420, 1990.

[Chui, 1992] Chui, C.K. *An Introduction to Wavelets*. Academic Press, New York, 1992.

[Class et al., 1993] Class, F., Kaltenmeier, A., Regel-Brietzmann, P. Optimization of an HMM-based continuous speech recognizer. In *Proc. European Conference on Speech Communication and Technology*, S. 803–806. Berlin, Germany, 1993.

[Clausi und Jernigan, 2000] Clausi, D.A., Jernigan, M.E. Designing Gabor filters for optimum texture separability. *Pattern Recognition*, 33:1835–1849, 2000.

[Cohen et al., 1992] Cohen, A., Daubechies, I., Feauveau, J.-C. Biorthogonal bases of compactly supported wavelets. *Comm. Pure and Applied Mathematics*, XLV:485–560, 1992.

[Coifman und Wickerhauser, 1992] Coifman, R., Wickerhauser, M.V. Entropy-based algorithms for best basis selection. *IEEE Trans. on Information Theory*, 38:713–718, 1992.

[Comaniciu et al., 2003] Comaniciu, D., Ramesh, V., Meer, P. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:564–577, 2003.

[Common, 1994] Common, P. Independent component analysis – a new concept? *Signal Processing*, 36(3):287–314, 1994.

[Cootes et al., 1993] Cootes, T.F., Taylor, C.J., Lanitis, A., Cooper, D.H., Graham, J. Building and using flexible models incorporating grey-level information. In *Proc. Intern. Conference on Computer Vision*, S. 242–246, 1993.

[Courant und Hilbert, 1953] Courant, R., Hilbert, D. *Methods of Mathematical Physics, Vol. I*. Interscience Publishers, New York, 1953.

[Cover, 1974] Cover, T.M. The best two independent measurements are not the two best. *IEEE Trans. on Systems, Man, and Cybernetics*, 4:116–117, 1974.

[Cover und Van Campenhout, 1977] Cover, T.M., Van Campenhout, J.M. On the possible orderings in the measurement selection problem. *IEEE Trans. on Systems, Man, and Cybernetics*, 7:657–661, 1977.

[Dash und Liu, 1997] Dash, M., Liu, H. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.

[Daubechies, 1988] Daubechies, I. Orthonormal bases of compactly supported wavelets. *Comm. Pure and Applied Mathematics*, XLI:909–996, 1988.

[Daubechies, 1992] Daubechies, I. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[Daubechies, 1993] Daubechies, I. Orthonormal bases of compactly supported wavelets II: Variations on a theme. *SIAM Journal of Mathematical Analysis*, 24(2):499–519, 1993.

[Daugman, 1985] Daugman, J. Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. *Journal Optical Society of America A*, 2:1160–1169, 1985.

[Davis und Mermelstein, 1980] Davis, S.B., Mermelstein, P. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 28:357–366, 1980.

[de Figueiredo, 1974] de Figueiredo, R.J.P. Optimal linear and nonlinear feature extraction based on the minimization of the increased risk of misclassification. ICSA-Report 275-025-014, Rice University, 1974.

[de Figueiredo, 1976] de Figueiredo, R.J.P. An algorithm for extraction of more than one optimal linear feature from several Gaussian pattern classes. In *Proc. 3. Int. Joint Conf. on Pattern Recognition*, S. 793–797. Coronado, Calif., 1976.

[Decell und Quirein, 1973] Decell, H.P., Quirein, J.A. An iterative approach to the feature selection problem. In *Proc. Machine Processing of Remotely Sensed Data*, S. 3 B1–3 B12. West Lafayette, USA, 1973.

[Deguchi und Morishita, 1978] Deguchi, K., Morishita, I. Texture characterization and texture based image partitioning using two-dimensional linear estimation techniques. *IEEE Trans. on Computers*, 27:739–745, 1978.

[Devijver, 1974] Devijver, P.A. On a new class of bounds on Bayes risk in multihypothesis pattern recognition. *IEEE Trans. on Computers*, 23:70–80, 1974.

[Do und Vetterli, 2002] Do, M.N., Vetterli, M. Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models. *IEEE Trans. on Multimedia*, S. 517–524, 2002.

[Docef et al., 2002] Docef, A., Kossentini, F., Nguuyen-Phi, K., Ismaeil, I.R. The quantized DCT and its application to DCT-based video coding. *IEEE Trans. on Image Processing*, 11:177–187, 2002.

[Dudani et al., 1977] Dudani, S.A., Breeding, K.J., McGhee, R.B. Aircraft identification by moment invariants. *IEEE Trans. on Computers*, 26:39–46, 1977.

[Duffin und Schaeffer, 1952] Duffin, R.J., Schaeffer, A.C. A class of nonharmonic Fourier series. *Trans. American Mathematical Society*, 72:341–366, 1952.

[Dunn und Higgins, 1995] Dunn, D., Higgins, W.E. Optimal Gabor filters for texture segmentation. *IEEE Trans. on Image Processing*, 4:947–964, 1995.

[Dunn et al., 1994] Dunn, D., Higgins, W.E., Wakeley, J. Texture segmentation using 2-D Gabor elementary functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:130–149, 1994.

[Eriksson und Koivunen, 2003] Eriksson, J., Koivunen, V. Characteristic-function-based independent component analysis. *Signal Processing*, 83:2195–2208, 2003.

[Fant, 1960] Fant, G. *The Acoustic Theory of Speech Production*. Mouton & Co., The Hague, The Netherlands, 1960.

[Figueiredo und Nowak, 2001] Figueiredo, M.A.T., Nowak, R.D. Wavelet-based image estimation: An empirical Bayes approach using Jeffrey's noninformative prior. *IEEE Trans. on Image Processing*, 10:1322–1331, 2001.

[Flanagan, 1972] Flanagan, J.L. *Speech Analysis, Synthesis, and Perception*, Bd. 3 von *Kommunikation und Kybernetik in Einzeldarstellungen*. Springer, Berlin, Heidelberg, New York, 2. Aufl., 1972.

[Flanagan, 1983] Flanagan, J.L. *Speech Analysis, Synthesis and Perception*. Springer, New York, 1983.

[Flusser, 2000] Flusser, J. Refined moment calculation using image block representation. *IEEE Trans. on Image Processing*, 9:1977–1978, 2000.

[Flusser et al., 2003] Flusser, J., Boldiš, J., Zitová, B. Moment forms invariant to rotation and blur in arbitrary number of dimensions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:234–246, 2003.

[Flusser und Suk, 1993] Flusser, J., Suk, T. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26:167–174, 1993.

[Fogel und Sagi, 1989] Fogel, I., Sagi, D. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–113, 1989.

[Fountain et al., 1998] Fountain, S.R., Tan, T.N., Baker, K.D. Comparative study of rotation invariant classification and retrieval of texture images. In *Proc. British Computer Vision Conference*, 1998.

[Freeman, 1978] Freeman, H. Shape description via the use of critical points. *Pattern Recognition*, 10:159–166, 1978.

[Freeman und Adelson, 1991] Freeman, W.T., Adelson, E.H. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[Fu, 1974] Fu, K.S. *Syntactic Methods in Pattern Recognition*. Academic Press, New York, 1974.

[Fu, 1982] Fu, K.S. *Syntactic Pattern Recognition and Applications*. Prentice Hall, Englewood Cliffs, N.J., 1982.

[Fu et al., 1970] Fu, K.S., Min, P.J., Li, T.J. Feature selection in pattern recognition. *IEEE Trans. on Systems Science and Cybernetics*, 6:33–39, 1970.

[Fukunaga, 1990] Fukunaga, K. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2. Aufl., 1990.

[Gabor, 1946] Gabor, D. Theory of communication. *The Journal of the Inst. of Electrical Engineers*, 93:429–457, 1946.

[Galloway, 1975] Galloway, M.M. Texture analysis using gray level run lenghts. *Computer Graphics and Image Processing*, 4:172–179, 1975.

[Gantmacher, 1958] Gantmacher, F.R. *Matrizenrechnung Teil I*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1958.

[Garnett und Yau, 1977] Garnett, J.M., Yau, S.S. Nonparametric estimation of the Bayes error rate of feature extractors using ordered nearest neighbor sets. *IEEE Trans. on Computers*, 26:46–54, 1977.

[Ghurumuruhan und Prabhu, 2004] Ghurumuruhan, G., Prabhu, K.M.M. Fixed-point fast Hartley transform error analysis. *Signal Processing*, 84(8):1307–1321, 2004.

[Girolami, 2002] Girolami, M. Mercer kernel-based clustering in feature space. *IEEE Trans. on Neural Networks*, 13:780–784, 2002.

[Goertzel, 1958] Goertzel, G. An algorithm for the evaluation of finite trigonometric series. *Am. Math. Monthly*, 65:34–35, 1958.

[Gonzalez und Thomason, 1978] Gonzalez, R.C., Thomason, M.G. *Syntactic Pattern Recognition, an Introduction*. Addison-Wesley, Reading, Mass., 1978.

[Gopinath und Burrus, 1992] Gopinath, R.A., Burrus, C.S. Wavelets and filter banks. In C.K. Chui, Hg., *Wavelets: A Tutorial in Theory and Applications*, S. 603–654. Academic Press, San Diego, CA, 1992.

[Gopinath und Burrus, 1995] Gopinath, R.A., Burrus, C.S. On cosine-modulated wavelet orthonormal bases. *IEEE Trans. on Image Processing*, 4:162–176, 1995.

[Götze et al., 1999] Götze, N., Drüe, S., Hartmann, G. Invariante Objekterkennung mit lokaler Fast-Fourier Mellin Transformation. In W. Förstner, J.M. Buhmann, A. Faber, P. Faber, Hg., *Mustererkennung 1999*, S. 155–163. Springer (Berlin, Heidelberg), 21. DAGM Symposium, Bonn, 1999.

[Govindaraju, 1996] Govindaraju, V. Locating human faces in photographs. *Int. Journal of Computer Vision*, 19(2):129–146, 1996.

[Granlund, 1972] Granlund, G.H. Fourier preprocessing for hand print character recognition. *IEEE Trans. on Computers*, 21:195–201, 1972.

[Gray und Goodman, 1995] Gray, R.M., Goodman, J.W. *Fourier Transforms, An Introduction for Engineers*. Kluwer Academic Publishers, Boston, MA, USA, 1995.

[Grbic et al., 2001] Grbic, N., Tao, X.-J., Nordholm, S.E., Claesson, I. Blind signal separation using overcomplete subband representation. *IEEE Trans. on Speech and Audio Processing*, 9(5):524–533, 2001.

[Grigorescu et al., 2002] Grigorescu, S.E., Petkov, N., Kruizinga, P. Comparison of texture features based on Gabor filters. *IEEE Trans. on Image Processing*, 11:1160–1167, 2002.

[Grossman, 1986] Grossman, A. Wavelet transform and edge detection. In M. Hazewinkel, Hg., *Stochastic Processes in Physics and Engineering*. Reidel, Dordrecht, 1986.

[Grossmann und Morlet, 1984] Grossmann, A., Morlet, J. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. of Math. Anal.*, 15:723–736, 1984.

[Gubner und Chang, 1995] Gubner, J.A., Chang, W.-B. Wavelet transforms for discrete-time periodic signals. *Signal Processing*, 42:167–180, 1995.

[Guillamet et al., 2003] Guillamet, D., Vitrià, J., Schiele, B. Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recognition Letters*, 24:2447–2454, 2003.

[Gülmezoğlu et al., 2001] Gülmezoğlu, M.B., Dzhafarov, V., Barkana, A. The common vector approach and its relation to pricipal component analysis. *IEEE Trans. on Speech and Audio Processing*, 9(6):655–662, 2001.

[Haar, 1910] Haar, A. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69:331–371, 1910.

[Haley und Manjunath, 1999] Haley, G.M., Manjunath, B.S. Rotation-invariant texture classification using complete space-frequency model. *IEEE Trans. on Image Processing*, 8:255–269, 1999.

[Haralick, 1979] Haralick, R.M. Statistical and structural approaches to texture. *Proc. IEEE*, 67:786–804, 1979.

[Haralick et al., 1973] Haralick, R.M., Shanmugan, K., Dinstein, I. Textural features for image classification. *IEEE Trans. Systems, Man, and Cybernetics,*, 3:610–621, 1973.

[Harbeck, 1996] Harbeck, M. *Objektorientierte linienbasierte Segmentierung von Bildern.* (Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg). Berichte aus der Informatik. Shaker Verlag, Aachen, Germany, 1996.

[Hariharan et al., 2001] Hariharan, R., Kiss, I., Viikki, O. Noise robust speech parameterization using multiresoulution feature extraction. *IEEE Trans. on Speech and Audio Processing*, 9:856–865, 2001.

[Harmuth, 1970] Harmuth, H.F. *Transmission of Information by Orthogonal Functions*. Springer, Berlin, 1970.

[Hermansky et al., 1985] Hermansky, H., Hanson, B., Wakita, H. Perceptually based linear predictive analysis of speech. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Bd. 2, S. 509–512. Tampa, Florida, 1985.

[Hermansky et al., 1986] Hermansky, H., Tsuga, K., Makino, S., Wakita, H. Perceptually based processing in automatic speech recognition. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 1971–1974. Tokyo, Japan, 1986.

[Hernando und Nadeu, 1991] Hernando, J., Nadeu, C. A comparative study of parameters and distances for noisy speech recognition. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Bd. 1, S. 91–94. Toronto, 1991.

[Hernando und Nadeu, 1997] Hernando, J., Nadeu, C. Linear prediction of the one-sided autocorrelation sequence for noisy speech recognition. *IEEE Trans. on Speech and Audio Processing*, 5:80–84, 1997.

[Hernando et al., 1997] Hernando, J., Nadeu, C., Marino, J. Speech recognition in a noisy car environment based on LP of the one-sided autocorrelation sequence and robust similarity measuring techniques. *Speech Communication*, 21:17–31, 1997.

[Heydorn, 1971] Heydorn, R.P. Redundancy in feature extraction. *IEEE Trans. on Computers*, 20:1051–1054, 1971.

[Hjaltason und Samet, 2003] Hjaltason, G.R., Samet, H. Properties of embedding methods for similarity searches in metric spaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:530–549, 2003.

[Hornegger et al., 2000] Hornegger, J., Niemann, H., Risack, R. Appearance-based object recognition using optimal feature transforms. *Pattern Recognition*, 33:209–224, 2000.

[Horst, 1979] Horst, R. *Nichtlineare Optimierung*, Kap. 3.4.1. Hanser Verlag, München, Wien, 1979.

[Hu, 1962] Hu, M.K. Visual pattern recognition by moment invariants. *IEEE Trans. on Information Theory*, 18:179–187, 1962.

[Huang und Chen, 1995] Huang, C.-L., Chen, Y.-T. Motion estimation method using a 3D-steerable filter. *Image and Vision Computing*, 13:21–32, 1995.

[Hummel, 1979] Hummel, R.A. Feature detection using basis functions. *Computer Graphics and Image Processing*, 9:40–55, 1979.

[Hyvärinen, 1999] Hyvärinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10:626–634, 1999.

[Hyvärinen, 2001] Hyvärinen, A. Blind source separation by nonstationarity of variance: A cumulant based approach. *IEEE Trans. on Neural Networks*, 12:1471–1474, 2001.

[Hyvärinen et al., 2001] Hyvärinen, A., Karhunen, J., Oja, E. *Independent Component Analysis*. J. Wiley, New York, USA, 2001.

[Hyvärinen und Oja, 1997] Hyvärinen, A., Oja, E. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[Iannarilli und Rubin, 2003] Iannarilli, F.J., Rubin, P.A. Feature selection for multiclass discrimination via mixed-integer linear programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:779–783, 2003.

[Ichino, 1981] Ichino, M. Nonparametric feature selection method based on local interclass structure. *IEEE Trans. on Systems, Man, and Cybernetics*, 10:289–296, 1981.

[Itakura und Saito, 1970] Itakura, F., Saito, S. A statistical method for estimation of speech spectral densities and formant frequencies. *El. and Comm. in Japan*, 53-A(1):36–43, 1970.

[Jacob und Unser, 2004] Jacob, M., Unser, M. Design of steerable filters for feature detection using Canny-like criteria. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1007–1019, 2004.

[Jain und Zongker, 1997] Jain, A., Zongker, D. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19:153–158, 1997.

[Jain und Dubes, 1978] Jain, A.K., Dubes, A. Feature definition in pattern recognition with small sample size. *Pattern Recognition*, 10:85–97, 1978.

[Jain und Farrokhnia, 1991] Jain, A.K., Farrokhnia, F. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.

[Jain und Karu, 1996] Jain, A.K., Karu, K. Learning texture discrimination masks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:195–205, 1996.

[Jankowski et al., 1995] Jankowski, C., Vo, H., Lippmann, R. A comparison of signal processing front ends for automatic word recognition. *IEEE Trans. on Speech and Audio Processing*, 3(4):286–293, 1995.

[Jäpel, 1980] Jäpel, D. Klassifikatorbezogene Merkmalsauswahl. Arbeitsberichte des IMMD Band 13, Nr. 4, Universität Erlangen-Nürnberg, Erlangen, 1980.

[Jiang und Bunke, 1991] Jiang, X.Y., Bunke, H. Simple and fast computation of moments. *Pattern Recognition*, 24:801–806, 1991.

[Jin et al., 2004] Jin, A.T.B., Ling, D.N.C., Song, O.T. An efficient fingerprint verification system using integrated wavelet and Fourier-Mellin invariant transform. *Image and Vision Computing*, 22:503–513, 2004.

[Johnston, 1980] Johnston, J.D. A filter family designed for use in quadrature mirror filter banks. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 291–294, 1980.

[Julesz, 1975] Julesz, B. Experiments in the visual perception of texture. *Scientific American*, 232(4):34–43, 1975.

[Julesz, 1982] Julesz, B. The role of terminators in preattentive preception of line textures. In D.G. Albrecht, Hg., *Recognition of Pattern and Form*, S. 33–55. Springer, Berlin, 1982.

[Julesz, 1986] Julesz, B. Texton gradients: The texton theory revisited. *Biological Cybernetics*, 54:247–251, 1986.

[Julesz und Bergen, 1983] Julesz, B., Bergen, R. Textons, the fundamental elements in preattentive vision and perception of textures. *Bell Systems Technical Journal*, 62(6):1619–1645, 1983.

[Kailath, 1967] Kailath, T. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Trans. on Communications*, 15:52–60, 1967.

[Kaiser, 1994] Kaiser, G. *A Friendly Guide to Wavelets*. Birkhäuser, Basel, 1994.

[Kaiser, 1990] Kaiser, J.F. On a simple algorithm to calculate the "energy" of a signal. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 381–384. Albuquerque, USA, 1990.

[Kaiser, 1993] Kaiser, J.F. Some useful properties of Teager's energy operator. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 149–152. Minneapolis, USA, 1993.

[Kan und Srinath, 2002] Kan, C., Srinath, M.D. Invariant character recognition with Zernicke and orthogonal Fourier-Mellin moments. *Pattern Recognition*, 35:143–154, 2002.

[Karhunen, 1946] Karhunen, K. Zur Spektraltheorie stochastischer Prozesse. *Ann. Acad. Sci. Fennicae*, (A)I, no. 34, 1946.

[Karnjanadecha und Zahorian, 2001] Karnjanadecha, M., Zahorian, S.A. Signal modelig for high-performance robust isolated word recognition. *IEEE Trans. on Speech and Audio Processing*, 9(6):647–654, 2001.

[Kashyap und Khotanzed, 1986] Kashyap, R.L., Khotanzed, A. A model-based method for rotation invariant texture classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:472–481, 1986.

[Kasprzak und Cichocki, 1996] Kasprzak, W., Cichocki, A. Hidden image separation from incomplete image mixtures by independent component analysis. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. Vol. II, 394–398. (IEEE Computer Society Press, Los Alamitos, USA), Wien, Austria, 1996.

[Kauppinen et al., 1995] Kauppinen, H., Seppänen, T., Pietikäinen, M. An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:201–207, 1995.

[Keidel und Neff, 1974, 1975, 1976] Keidel, W.D., Neff, W.D. *Auditory System, Handbook of Sensory Physiology*, Bd. VI/1-3. Springer, Berlin, Heidelberg, New York, 1974, 1975, 1976.

[Keller et al., 1989] Keller, J., Crownover, R., Chen, S. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and Image Processing*, 45:150–160, 1989.

[Keren et al., 1994] Keren, D., Cooper, D., Subrahmonia, J. Describing complicated obejects by implicit polynomials. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:38–53, 1994.

[Keysers et al., 2000] Keysers, D., Dahmen, J., Theiner, T., Ney, H. Experiments with an extended tangent distance. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. Vol. 2, 38–42. Barcelona, Spain, 2000.

[Keysers et al., 2004] Keysers, D., Macherey, W., Ney, H., Dahmen, J. Adaptation and statistical pattern recognition using tangent vectors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:269–274, 2004.

[Khalil und Bayoumi, 2001] Khalil, M.I., Bayoumi, M.M. A dyadic wavelet affine invariant function for 2D shape recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:1152–1164, 2001.

[Khalil und Bayoumi, 2002] Khalil, M.I., Bayoumi, M.M. Affine invariants for object recognition using the wavelet transform. *Pattern Recognition Letters*, 23(1-3):57–72, 2002.

[Khotanzad, 1990] Khotanzad, A. Invariant image recognition by Zernike moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:489–497, 1990.

[Kim et al., 2002a] Kim, H.-C., Kim, D., Bang, S.Y. A numeral character recognition using the PCA mixture model. *Pattern Recognition Letters*, 23:103–111, 2002a.

[Kim et al., 2002b] Kim, K.I., Jung, K., Park, S.H., Kim, H.J. Support vector machines for texture classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1542–1550, 2002b.

[Kim und Udpa, 2000] Kim, N.D., Udpa, S. Texture classification using rotated wavelet filters. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 30:847–852, 2000.

[Kingsbury, 1998] Kingsbury, N.G. The dual-tree complex wavelet transform: A new efficient tool for image restoration and enhancement. In *Proc. European Signal Processing Conference*, S. 319–322, 1998.

[Kirby und Sirovich, 1990] Kirby, M., Sirovich, L. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:103–108, 1990.

[Kittler, 1975] Kittler, J. Mathematical methods of feature selection in pattern recognition. *Int. Journal Man-Machine Studies*, 7:609–637, 1975.

[Kittler, 1978] Kittler, J. Feature set search algorithms. In C.H. Chen, Hg., *Pattern Recognition and Signal Processing (Proceedings of the NATO Advanced Study)*, S. 41–60. Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1978.

[Kleinert, 2004] Kleinert, F. *Wavelets and Multiwavelets*. Studies in Advanced Mathematics. Chapman & Hall/CRC, Boca Raton, Florida, USA, 2004.

[Korbut und Finkelstein, 1971] Korbut, A.A., Finkelstein, J.J. *Diskrete Optimierung*, Kap. 10. Akademie Verlag, Berlin, 1971.

[Kramer, 1991] Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks. *Journ. of the American Institute of Chemical Engineers*, 37(2):233–243, 1991.

[Kudo und Sklansky, 2000] Kudo, M., Sklansky, J. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33:25–41, 2000.

[Kuhn et al., 2000] Kuhn, R., Junqua, J.-C., Nguyen, P., Niedzielski, N. Rapid speaker adaptation in eigenvoice space. *IEEE Trans. on Speech and Audio Processing*, 8(6):695–707, 2000.

[Kullback, 1968] Kullback, S. *Information Theory and Statistics*. Dover Publications, New York, USA, 1968.

[Kuthirummal et al., 2004] Kuthirummal, S., Jawahar, C.V., Narayanan, P.J. Fourier domain representation of planar curves for recognition in multiple views. *Pattern Recognition*, 37:739–754, 2004.

[Kwak und Choi, 2002a] Kwak, N., Choi, C.-H. Input feature selection by mutual information based on Parzen window. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1667–1671, 2002a.

[Kwak und Choi, 2002b] Kwak, N., Choi, C.-H. Input feature selection for classification problems. *IEEE Trans. on Neural Networks*, 13:143–159, 2002b.

[Kyrki et al., 2004] Kyrki, V., Kamarainen, J.-K., Kälviäinen, H. Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters*, 25:311–318, 2004.

[Lai et al., 2001] Lai, J.H., Yuen, P.C., Feng, G.C. Face recognition using holistic Fourier invariant features. *Pattern Recognition*, 34:95–109, 2001.

[Laine und Fan, 1993] Laine, A., Fan, J. Texture classification by wavelet packet signature. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:1186–1191, 1993.

[Lawler und Wood, 1966] Lawler, E., Wood, D. Branch and bound methods: A survey. *Operations Res.*, 14:699–719, 1966.

[Laws, 1980] Laws, K.L. *Texture Image Classification*. Dissertation, University of Southern California, Faculty of the Graduate School, Los Angeles, 1980.

[Le Borgne et al., 2004] Le Borgne, H., Guérin-Dugué, A., Antoniadis, A. Representation of images for classification with independent features. *Pattern Recognition Letters*, 25:141–151, 2004.

[Ledley, 1964] Ledley, R.S. High-speed automatic analysis of biomedical pictures. *Science*, 146:216–223, 1964.

[Ledley, 1972] Ledley, R.S. Analysis of cells. *IEEE Trans. on Computers*, 21:740–752, 1972.

[Lee et al., 2003] Lee, C., Hyun, D., Choi, E., Go, J., Lee, C. Optimizing feature extraction for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 11:80–87, 2003.

[Lee und Seung, 1999] Lee, D.D., Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[Lee und Lewicki, 2002] Lee, T.-W., Lewicki, M.S. Unsupervised image classification, segmentation,

and enhancement using ICA mixture models. *IEEE Trans. on Image Processing*, 11:270–279, 2002.

[Leonardis und Bischof, 1996] Leonardis, A., Bischof, H. Robust recovery of eigenimages in the presence of outliers and occlusions. *Journal of Computing and Information Technology*, 4(1):25–38, 1996.

[Lerner und Mazor, 1992] Lerner, S., Mazor, B. Telephone channel normalization for automatic speech recognition. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Bd. 1, S. 261–264. San Francisco, 1992.

[Levinson, 1947] Levinson, N. The Wiener RMS (root mean square) error criterion in filter design and prediction. *J. Mathematical Physics*, 25:261–278, 1947.

[Lewis, 1962] Lewis, P.M. The characteristic selection problem in recognition systems. *IEEE Trans. on Information Theory*, 8:171–178, 1962.

[Li, 1991] Li, B. Fast computation of moment invariants. *Pattern Recognition*, 24:807–813, 1991.

[Li, 1993] Li, B. The moment calculation of polyhedra. *Pattern Recognition*, 26:1229–1233, 1993.

[Li et al., 2005] Li, H., Liu, G., Zhang, Z. Optimization of integer wavelet transforms based on difference correlation structures. *IEEE Trans. on Image Processing*, 14:1831–1847, 2005.

[Li und Leung, 2002] Li, L., Leung, M.K.H. Integrating intensity and texture differences for robust change detection. *IEEE Trans. on Image Processing*, 11:105–112, 2002.

[Li und Yuan, 2005] Li, M., Yuan, B. 2D-LDA: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26:527–532, 2005.

[Li et al., 2000] Li, Q., Soong, F., Siohan, O. A high-performance auditory feature for robust speech recognition. In *Proc. Int. Conference on Spoken Language Processing*, 2000.

[Li et al., 1996] Li, W., Haese-Coat, V., Ronsin, J. Using adaptive genetic algorithms in the design of morphological filters in textural image processing. In *Nonlinear Image Processing VII*, S. 24–35. SPIE, 1996.

[Liggett, 1988] Liggett, J.A. Exact formulae for areas, volumes, and moments of polygons and polyhedra. *Comm. Applied Numerical Mathematics*, 4(6):815–820, 1988.

[Lim, 1979] Lim, J.S. Spectral root homomorphic deconvolution system. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 27(3):223–233, 1979.

[Lissack und Fu, 1976] Lissack, T., Fu, K.S. Error estimation in pattern recognition via L^{α}-distance between posterior density functions. *IEEE Trans. on Information Theory*, 22:34–45, 1976.

[Liu und Wechsler, 2003] Liu, C., Wechsler, H. Independent component analysis of Gabor features for face recognition. *IEEE Trans. on Image Processing*, 14:919–928, 2003.

[Liu und Wang, 2003] Liu, X., Wang, D.L. Texture classification using spectral histograms. *IEEE Trans. on Image Processing*, 12:661–670, 2003.

[Loève, 1955] Loève, M. *Probability Theory*. Van Nostrand, Princeton, N.J., USA, 1955.

[Lonnestad, 1992] Lonnestad, T. A new set of texture features based on the Haar transform. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. Vol. III, 676–679. The Hague, The Netherlands, 1992.

[Lowe, 2004] Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60:91–110, 2004.

[Lung, 2004a] Lung, S.-Y. Feature extracted from wavelet eigenfunction estimation for text independent speaker recognition. *Pattern Recognition*, 37:1543–1544, 2004a.

[Lung, 2004b] Lung, S.-Y. Further reduced form of wavelet feature for text independent speaker recognition. *Pattern Recognition*, 37:1569–1570, 2004b.

[Magarey und Kingsbury, 1998] Magarey, J., Kingsbury, N. Motion estimation using a complex valued wavelet transform. *IEEE Trans. on Signal Processing*, 46:1069–1084, 1998.

[Mak et al., 2004] Mak, B.K.-W., Tam, Y.-C., Li, P.Q. Discriminative auditory-based features for robust speech recognition. *IEEE Trans. on Speech and Audio Processing*, 12:27–36, 2004.

[Makhoul, 1975] Makhoul, J. Linear prediction, a tutorial review. *Proc. IEEE*, 63:561–580, 1975.

[Mallat, 1989] Mallat, S. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[Mallat und Hwang, 1992] Mallat, S., Hwang, W.L. Singularity detection and processing with wavelets. *IEEE Trans. on Information Theory*, 38:617–643, 1992.

[Mallat und Zhong, 1992] Mallat, S., Zhong, S. Characterization of signals from muliscale edges. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(7):710–732, 1992.

[Malthouse, 1998] Malthouse, E.C. Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Trans. on Neural Networks*, 9:165–173, 1998.

[Mansour und Juang, 1989] Mansour, D., Juang, B. The short-time modified coherence representation and noisy speech recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37(6):795–804, 1989.

[Manthalkar et al., 2003a] Manthalkar, R., Biswas, P.K., Chatterji, B.N. Rotation and scale invariant texture features using discrete wavelet packet transform. *Pattern Recognition Letters*, 24:2455–2462, 2003a.

[Manthalkar et al., 2003b] Manthalkar, R., Biswas, P.K., Chatterji, B.N. Rotation invariant texture classification using even symmetric Gabor filters. *Pattern Recognition Letters*, 24:2061–2068, 2003b.

[Markel und Gray Jr., 1976] Markel, J.D., Gray Jr., A.H. *Linear Prediction of Speech*, Bd. 12 von *Communications and Cybernetics*. Springer, Berlin, Heidelberg, 1976.

[Markel und Gray Jr., 1982] Markel, J.D., Gray Jr., A.H. *Linear Prediction of Speech*, Bd. 12 von *Communications and Cybernetics*. Springer, Berlin, Heidelberg, 3. Aufl., 1982.

[Marril und Green, 1963] Marril, T., Green, D.M. On the effectiveness of receptors in recognition systems. *IEEE Trans. on Information Theory*, 9:11–17, 1963.

[Martinez und Kak, 2001] Martinez, A.M., Kak, A.C. PCA versus LDA. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:228–233, 2001.

[Matsakis et al., 2004] Matsakis, P., Keller, J.M., Sjahputera, O., Marjamaa, J. The use of forc histograms for affine-invariant relative position description. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1–18, 2004.

[Matusita, 1966] Matusita, K. A distance and related statistics in multivariate analysis. In P.R. Krishnaiah, Hg., *Multivariate Analysis*, S. 178–200. Academic Press, New York, 1966.

[McGinn und Johnson, 1983] McGinn, D., Johnson, D. Reduction of all-pole parameter estimator bias by successive autocorrelation. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Bd. 3, S. 1088–1091. Boston, 1983.

[Mel, 1997] Mel, B.W. SEEMORE: Combining color, shape, and texture histogramming in neurally inspired approach to visual object recognition. *Neural Computation*, 9(4):777–804, 1997.

[Meyer, 1993] Meyer, Y. *Wavelets - Algorithms and Applications*. SIAM, Philadelphia, 1993.

[Middleton, 1960] Middleton, D. *An Introduction to Statistical Communication Theory*. McGraw Hill, New York, 1960.

[Mika et al., 1999] Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.-R. Fisher discriminant analysis with kernels. In Y.H. Hu, J. Larsen, E. Wilson, S. Douglas, Hg., *Proc. Neural Networks for Signal Processing*, S. 41–48. IEEE, 1999.

[Mika et al., 2003] Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., Müller, K.R. Constructing descriptive and discriminative nonlinear features; Rayleigh coefficients in kernel feature spaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:623–628, 2003.

[Mindru et al., 1999] Mindru, F., Moons, T., van Gool, L. Recognizing color patterns irrespective of viewpoint and illumination. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, S. Vol. 1, 368–373, 1999.

[Mitra et al., 2002] Mitra, P., Murthy, C.A., Pal, S.K. Unsupervised feature selection using feature similarity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:301–312, 2002.

[Moayer und Fu, 1975] Moayer, B., Fu, K.S. A syntactic approach to fingerprint pattern recognition. *Pattern Recognition*, 7:1–23, 1975.

[Moayer und Fu, 1976] Moayer, B., Fu, K.S. A tree system approach for fingerprint pattern recognition. *IEEE Trans. on Computers*, 25:262–274, 1976.

[Molgedy und Schuster, 1994] Molgedy, L., Schuster, H.G. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634–3637, 1994.

[Mori et al., 1970] Mori, K., Genchi, H., Watanabe, S., Katsuragi, S. Microprogram controlled pattern processing in a handwritten mail reader-sorter. *Pattern Recognition*, 2:175–185, 1970.

[Mucciardi und Gose, 1971] Mucciardi, A.N., Gose, E.E. A comparison of seven techniques for choosing subsets of pattern recognition properties. *IEEE Trans. on Computers*, 20:1023–1031, 1971.

[Mukandan et al., 2001] Mukandan, R., Ong, S.H., Lee, P.A. Image analysis by Tchebichev moments. *IEEE Trans. on Image Processing*, 10(9):1357–1364, 2001.

[Mukundan und Ramakrishnan, 1995] Mukundan, R., Ramakrishnan, K.R. Fast computation of Legendre and Zernike moments. *Pattern Recognition*, 28:1433–1442, 1995.

[Müller et al., 2001] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12:181–201, 2001.

[Mundy und Zisserman, 1992] Mundy, J.L., Zisserman, A. *Geometric Invariants in Computer Vision*. The MIT Press, Cambridge MA USA, 1992.

[Muñoz et al., 2002] Muñoz, A., Ertlé, R., Unser, M. Continuous wavelet transform with arbitrary scales and $O(N)$ complexity. *Signal Processing*, 82:749–757, 2002.

[Murase und Lindenbaum, 1995] Murase, H., Lindenbaum, M. Spatial temporal adaptive method for partial eigenstructure decomposition of large matrices. *IEEE Trans. on Image Processing*, 4:620–629, 1995.

[Murase und Nayar, 1995] Murase, H., Nayar, S. Visual learning and recognition of 3D objects from appearance. *Int. Journal of Computer Vision*, 14:5–24, 1995.

[Narendra und Fukunaga, 1977] Narendra, P., Fukunaga, K. A branch and bound algorithm for feature subset selection. *IEEE Trans. on Computers*, 26:917–922, 1977.

[Niemann, 1969] Niemann, H. *Begründung und Anwendung einer Theorie zur quantitativen Beschreibung und Erkennung von Mustern*. Dissertation, Techn. Universität Hannover, Hannover, 1969.

[Niemann, 1970] Niemann, H. Mustererkennung mit orthonormalen Reihenentwicklungen. *Nachrichtentechn. Zeitschrift*, 23:308–313, 1970.

[Niemann, 1971] Niemann, H. An improved series expansion for pattern recognition. *Nachrichtentechn. Zeitschrift*, 24:473–477, 1971.

[Niemann, 1974] Niemann, H. *Methoden der Mustererkennung*. Akademische Verlagsgesellschaft, Frankfurt, 1974.

[Niemann, 1981] Niemann, H. *Pattern Analysis*. Springer Series in Information Sciences 4. Springer, Berlin, Heidelberg, New York, 1981.

[Niemann, 1983] Niemann, H. *Klassifikation von Mustern*. Springer; (zweite erweiterte Auflage 2003 im Internet: http://www5.informatik.uni-erlangen.de/MEDIA/nm/klassifikation-von-mustern/m00links.html), Berlin, Heidelberg, 1983.

[Niemann et al., 1988] Niemann, H., Brietzmann, A., Ehrlich, U., Posch, S., Regel, P., Sagerer, G., Salzbrunn, R., Schukat-Talamazzini, G. A knowledge based speech understanding system. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2:321–350, 1988.

[Niemann und Winkler, 1969] Niemann, H., Winkler, G. Eine Theorie zur quantitativen Beschreibung und Erkennung von Mustern. *Nachrichtentechn. Zeitschrift*, 22:94–100, 1969.

[Nussbaumer, 1981] Nussbaumer, H.J. *Fast Fourier Transform and Convolution Algorithms*. Springer, Berlin, Heidelberg, New York, 1981.

[Oirrak et al., 2002] Oirrak, A.E., Daoudi, M., Aboutajdine, D. Affine invariant descriptors using Fourier series. *Pattern Recognition Letters*, 23:1109–1118, 2002.

[Oirrak et al., 2003] Oirrak, A.E., Daoudi, M., Aboutajdine, D. Affine invariant descriptors for color images using Fourier series. *Pattern Recognition Letters*, 24:1339–1348, 2003.

[Oja, 1982] Oja, E. A simplified neuron model as a pricipal component analyzer. *Journal of Mathema-*

tical Biology, 15:267–273, 1982.

[Oja, 1983] Oja, E. *Subspace Methods of Pattern Recognition*. Research Studies Press, Letchworth, UK, 1983.

[Oja, 1992] Oja, E. Principal components, minor components and linear neural networks. *Neural Networks*, 5:927–935, 1992.

[Oja et al., 2004] Oja, E., Harmeling, S., Almeida, L. Special section independent component analysis and beyond. *Signal Processing*, 84(2), 2004.

[Oppenheim, 1968] Oppenheim, A.V. Nonlinear filtering of multiplied and convolved signals. *Proc. IEEE*, 56:1264–1291, 1968.

[Oppenheim und Schafer, 1975] Oppenheim, A.V., Schafer, R.W. *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1975.

[Ozeki, 1979] Ozeki, K. A coordinate-free theory of eigenvalue analysis related to the method of principal components. *Information and Control*, 42:38–59, 1979.

[Panda und Chatterji, 1997] Panda, R., Chatterji, B.N. Unsupervised texture segmentation using tuned filters in Gaborian space. *Pattern Recognition Letters*, 18:445–453, 1997.

[Park et al., 2004] Park, S.B., Lee, J.W., Kim, S.K. Content-based image classification using a neural network. *Pattern Recognition Letters*, 25:287–300, 2004.

[Parzen, 1974] Parzen, E. Some recent advances in time series modeling. *IEEE Trans. on Automatic Control*, 19:723–730, 1974.

[Pastor et al., 2001] Pastor, L., Rodriguez, A., Espadero, J., Rincon, L. 3D wavelet-based multiresolution object representation. *Pattern Recognition*, 34:2497–2513, 2001.

[Patrick und Fisher, 1969] Patrick, E.A., Fisher, F.P. Nonparametric feature selection. *IEEE Trans. on Information Theory*, 15:577–584, 1969.

[Pattichis et al., 2001] Pattichis, M.S., Bovik, A.C., Havlicek, J.W., Sidiropoulos, N.D. Multidimensional orthogonal FM transform. *IEEE Trans. on Image Processing*, 10:448–464, 2001.

[Pavlidis und Ali, 1979] Pavlidis, T., Ali, F. A hierarchical syntactic shape analyzer. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:2–9, 1979.

[Pavlidis und Horowitz, 1974] Pavlidis, T., Horowitz, S.L. Segmentation of plane curves. *IEEE Trans. on Computers*, 23:860–870, 1974.

[Pease, 1968] Pease, M.C. An adaptation of the fast Fourier transform to parallel processing. *Journal of the Association for Comp. Machinery*, 15:252–264, 1968.

[Pei und Yeh, 1997] Pei, S.-C., Yeh, M.-H. An introduction to discrete finite frames. *IEEE Signal Processing Magazine*, 14(6):84–96, 1997.

[Peleg et al., 1984] Peleg, S., Naor, J., Hartley, R., Avnir, D. Multiresolution texture analysis and classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:518–523, 1984.

[Pentland, 1984] Pentland, A.P. Fractal based descripion of natural scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:661–674, 1984.

[Pérez-Rendón und Robles, 2004] Pérez-Rendón, A.F., Robles, R. The convolution theorem for the continuous wavelet-transform. *Signal Processing*, 84:55–67, 2004.

[Persoon und Fu, 1977] Persoon, E., Fu, K.S. Shape discrimination using Fourier descriptors. *IEEE Trans. on Systems, Man, and Cybernetics*, 7:170–179, 1977.

[Persoon und Fu, 1986] Persoon, E., Fu, K.S. Shape discrimination using Fourier descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:388–397, 1986.

[Petrou und Kadyrov, 2004] Petrou, M., Kadyrov, A. Affine invariant features from the trace transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:30–44, 2004.

[Pham, 2001] Pham, D.-T. Blind separation of instantaneous mixture of sources via the Gaussian mutual information criterion. *Signal Processing*, 81(4):855–870, 2001.

[Pichler et al., 1996] Pichler, O., Teuner, A., Hosticka, B.J. A comparison of texture feature extraction using adaptive Gabor filtering, pyramidal and tree structured wavelet transforms. *Pattern Recognition*, 29:733–742, 1996.

[Picone, 1993] Picone, J.W. Signal modeling techiques in speech recognition. *Proc. IEEE*, 81:1215–1247, 1993.

[Pikaz und Averbuch, 1997] Pikaz, A., Averbuch, A. An efficient topological characterization of gray-level textures using a multiresolution representation. *Graphical Models and Image Processing*, 59:1–17, 1997.

[Plannerer und Ruske, 1992] Plannerer, B., Ruske, G. Recognition of demisyllable based units using semicontinuous hidden Markov models. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Bd. 1, S. 581–584. San Francisco, USA, 1992.

[Plomp, 1964] Plomp, R. The ear as a frequency analyzer. *Journ. of the Acoustical Society of America*, 36:1628–1636, 1964.

[Plumbley, 2003] Plumbley, M.D. Algorithms for nonnegative independent component analysis. *IEEE Trans. on Neural Networks*, 14:534–543, 2003.

[Plumbley und Oja, 2004] Plumbley, M.D., Oja, E. A "Nonnegative PCA" algorithm for independent component analysis. *IEEE Trans. on Neural Networks*, 15:66–76, 2004.

[Pols, 1977] Pols, L.C.W. Spectral analysis and identification of Dutch vowels in monosyllabic words. Techn. Ber., University of Amsterdam, Amsterdam, The Neterlands, 1977.

[Pösl, 1999] Pösl, J. *Erscheinungsbasierte statistische Objekterkennung*. Berichte aus der Informatik. Shaker Verlag, Aachen, Germany, 1999.

[Pratt, 1991] Pratt, W.K. *Digital Image Processing*. Wiley-Interscience, New York, 2. Aufl., 1991.

[Press et al., 1994] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 2. Aufl., 1994.

[Press et al., 2002] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. *Numerical Recipes in C++. The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 2. Aufl., 2002.

[Prokop und Reeves, 1992] Prokop, R.J., Reeves, A.P. A survey of moment based techniques for unoccluded object recognition. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, 54(5):438–460, 1992.

[Pudil et al., 1994] Pudil, P., Novovičová, J., Kittler, J. Floating search methods for feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[Pun, 2003] Pun, C.-M. Rotation invariant texture features for image retrieval. *Computer Vision and Image Understanding*, 89(1):24–43, 2003.

[Pun und Lee, 2003] Pun, C.-M., Lee, M.-C. Log-polar wavelet energy signatures for rotation and scale invariant texture classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:590–603, 2003.

[Ramer, 1972] Ramer, U. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1:244–256, 1972.

[Randen und Husøy, 1999] Randen, T., Husøy, J.H. Filtering for texture classification: a comparative study. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:291–310, 1999.

[Reeves et al., 1988] Reeves, A.P., Prokop, R.J., Andrews, S.E., Kuhl, F.P. Three-dimensional shape analysis using moments and Fourier descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10:937–943, 1988.

[Regalia und Kofidis, 2003] Regalia, P.A., Kofidis, E. Monotonic covergence of fixed-point algorithms for ICA. *IEEE Trans. on Image Processing*, 14:943–949, 2003.

[Regel, 1988] Regel, P. *Akustisch-Phonetische Transkription für die automatische Spracherkennung*. Fortschrittberichte, Reihe 10 Nr. 83. VDI Verlag, Düsseldorf, Germany, 1988.

[Reinhardt, 1979] Reinhardt, K.H. Algorithmen zur Merkmalsauswahl mit Implementierung eines Verfahrens. Diplomarbeit, Lehrstuhl für Mustererkennung (Informatik 5), Univ. Erlangen-Nürnberg, Erlangen, 1979.

[Reinhold, 2003] Reinhold, M. *Robuste, probabilistische, erscheinungsbasierte Objekterkennung*. Dis-

sertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 2003.

[Reiss, 1993] Reiss, T.-H. *Recognizing Planar Objects Using Invariant Image Features*, Bd. 676 von *Lecture Notes in Computer Science*. Springer, Berlin, 1993.

[Reiss, 1991] Reiss, T.H. The revised fundamental theorem of moment invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:830–834, 1991.

[Reitboeck und Brody, 1969] Reitboeck, H., Brody, T.P. A transformation with invariance under cyclic permutation for applications in pattern recognition. *Information and Control*, 15:130–154, 1969.

[Ren et al., 2002] Ren, M., Yang, J., Sun, H. Tracing boundary contours in a binary image. *Image and Vision Computing*, 20:125–131, 2002.

[Rieck, 1995] Rieck, S. *Parametrisierung und Klassifikation gesprochener Sprache*, Bd. 353 von *Fortschrittberichte Reihe 10*. VDI Verlag, Düsseldorf, Germany, 1995.

[Rioul und Vetterli, 1991] Rioul, O., Vetterli, M. Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8(4):14–38, 1991.

[Rivoira und Torasso, 1978] Rivoira, S., Torasso, P. An isolated word recognizer based on grammar controlled classification processes. *Pattern Recognition*, 10:73–84, 1978.

[Rosenfeld, 1975] Rosenfeld, A. A characterization of parallel thinning algorithms. *Information and Control*, 29:286–291, 1975.

[Rosenfeld und Johnston, 1973] Rosenfeld, A., Johnston, E. Angle detection on digital curves. *IEEE Trans. on Computers*, 22:875–878, 1973.

[Rothe et al., 1996] Rothe, I., Suesse, H., Voss, K. The method of normalization to determine invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:366–377, 1996.

[Rothwell et al., 1992] Rothwell, C.A., Zisserman, A., Forsyth, D.A., Mundy, J.L. Canonical frames for planar object recognition. In *Proc. European Conference on Computer Vision (ECCV)*, S. 757–772. Springer, Santa Margherita Ligure, 1992.

[Rubner et al., 2001] Rubner, Y., Puzicha, J., Tomasi, C., Buhmann, J.M. Empirical evaluation of dissimilarity values for color and texture. *Computer Vision and Image Understanding*, 84:25–43, 2001.

[Rushing et al., 2001] Rushing, J.A., Ranganath, H.S., Hinke, T.H., Graves, S.J. Using association rules as texture features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:845–858, 2001.

[Ruske, 1979] Ruske, G. *Automatische Erkennung gesprochener Wörter mit einem Funktionsmodell des Gehörs*, Bd. 6 von *Nachrichten Elektronik*. Dr. Alfred Hüthig Verlag, Heidelberg, 1979.

[Ruske, 1994] Ruske, G. *Automatische Spracherkennung: Methoden der Klassifikation und Merkmalsextraktion*. Oldenbourg Verlag, München, 2. Aufl., 1994.

[Ruske und Schotola, 1978] Ruske, G., Schotola, T. An approach to speech recognition using syllabic decision units. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 722–725. Tulsa, Oklahoma, 1978.

[Ruske und Schotola, 1981] Ruske, G., Schotola, T. The efficiency of demisyllable segmentation in the recognition of spoken words. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 971–974. Atlanta, USA, 1981.

[Sadjadi und Hall, 1980] Sadjadi, F.A., Hall, E.L. Three-dimensional moment invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:127–136, 1980.

[Saito, 1994] Saito, N. *Local Feature Extraction and Its Applications Using a Library of Bases*. Dissertation, Department of Mathematics, Yale University, New Haven, CT, USA, 1994.

[Saito und Coifman, 1995] Saito, N., Coifman, R.R. Local discriminant bases and their applications. *Journal of Mathematical Imaging and Vision*, 5:337–358, 1995.

[Sanger, 1989] Sanger, T.D. Optimal unsupervised learning in a single-layer feedforward neural network. *Neural Networks*, 1:495–473, 1989.

[Sarikaya, 2001] Sarikaya, R. *Robust and Efficient Techniques for Speech Recognition*. Dissertation, Duke University, Dept. ECE, 2001.

[Sarikaya und Hansen, 2001] Sarikaya, R., Hansen, J.H.L. Analysis of the root-cepstrum for acoustic

modeling and fast decoding in speech recognition. In *Proc. European Conference on Speech Communication and Technology*, S. 687–690. Aalborg, Denmark, 2001.

[Sarkar und Chaudhuri, 1992] Sarkar, N., Chaudhuri, B.B. An efficient approach to estimate fractal dimenison of textural images. *Pattern Recognition*, 25:1035–1041, 1992.

[Saupe und Vranić, 2001] Saupe, D., Vranić, D.V. 3D model retrieval with spherical harmonics and moments. In B. Radig, S. Florczyk, Hg., *Pattern Recognition. Proc. 23rd DAGM Symposium*, S. 392–397. (Springer LNCS 2191, Berling Heidelberg, ISBN 3-540-42596-9), München, Germany, 2001.

[Schiele und Crowley, 1996] Schiele, B., Crowley, J.L. Object recognition using multidimensional receptive field histograms. In B.F. Buxton, R. Cipolla, Hg., *Proc. European Conference on Computer Vision (ECCV)*, S. Vol. 1, 610–619. (Springer LNCS 1064, Berlin Heidelberg), Cambridge, UK, 1996.

[Schiele und Crowley, 2000] Schiele, B., Crowley, J.L. Recognition without correspondence using multidimensional receptive field histograms. *Int. Journal of Computer Vision*, 36(1):31–52, 2000.

[Schiele und Pentland, 1999] Schiele, B., Pentland, A. Probabilistic object recognition and localization. In *Proc. Int. Conference on Computer Vision (ICCV)*, S. Vol. 1, 177–182. (IEEE Computer Society, Los Alamitos, California, ISBN 0-7695-0164-8), Kerkyra, Greece, 1999.

[Schless, 1999] Schless, V. *Automatische Erkennung von gestörten Sprachsignalen*. Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 1999.

[Schmid und Mohr, 1996] Schmid, C., Mohr, R. Combining greavalue invariants with local constraints for object recognition. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, S. 872–877. San Francisco, CA, USA, 1996.

[Schneeweiß, 1974] Schneeweiß. *Dynamisches Programmieren*. Physica Verlag, Würzburg, Wien, 1974.

[Schölkopf et al., 1999] Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.-R., Rätsch, G., Smola, A.J. Input space versus feature space in kernel-based methods. *IEEE Trans. on Neural Networks*, 10:1000–1017, 1999.

[Schotola, 1984] Schotola, T. On the use of demisyllables in automatic word recognition. *Speech Communication*, 3:63–87, 1984.

[Schukat-Talamazzini, 1995] Schukat-Talamazzini, E.G. *Automatische Spracherkennung*. Vieweg, Wiesbaden, 1995.

[Schulz-Mirbach, 1994] Schulz-Mirbach, H. Constructing invariant features by averaging techniques. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. Vol. II, 387–390. IEEE Computer Society Press, Jerusalem, 1994.

[Schulz-Mirbach, 1995] Schulz-Mirbach, H. *Anwendung von Invarianzprinzipien zur Merkmalgewinnung in der Mustererkennung*. Fortschrittberichte, Reihe 10. VDI Verlag, Düsseldorf, Germany, 1995.

[Schwartz und Austin, 1991] Schwartz, R., Austin, S. A comparison of several approximate algorithms for finding multiple (n-best) sentence hypotheses. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 701–704. Toronto, Canada, 1991.

[Seidel, 1974] Seidel, H. *Verarbeitungstechniken zur Datenreduktion bei Kurzzeitfrequenzspektren von Sprachsignalen*. Dissertation, Fachbereich Elektrotechnik, Technische Universität München, München, Germany, 1974.

[Seneff, 1986] Seneff, S. A computational model for the peripheral auditory system: Application to speech recognition research. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 1983–1986. Tokyo, Japan, 1986.

[Shaick und Yaroslavski, 2001] Shaick, B.-Z., Yaroslavski, L.P. Object localization using linear adaptive filters. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, Hg., *Vision, Modeling, and Visualization 2001 (Proceedings of the International Workshop, Stuttgart, Germany)*, S. 11–18. Akademische Verlagsgesellschaft, Berlin, Germany, 2001.

[Sheinberg, 1970] Sheinberg, I. The input 2 document reader (a new optical character recognition system). *Pattern Recognition*, 2:161–173, 1970.

[Shelman, 1972] Shelman, C.B. The application of list processing. *Pattern Recognition*, 4:201–210, 1972.

[Sheynin und Tuzikov, 2001] Sheynin, S.A., Tuzikov, A.V. Explicit formulae for polyhedra moments. *Pattern Recognition Letters*, 22:1103–1109, 2001.

[Shu et al., 2002] Shu, H.Z., Luo, L.M., Zhou, J.D., Bao, X.D. Moment-based methods for polygonal approximation of digitized curves. *Pattern Recognition*, 35:421–434, 2002.

[Siedlecki und Sklansky, 1988] Siedlecki, W., Sklansky, J. On automatic feature selection. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[Simoncelli und Farid, 1996] Simoncelli, E.P., Farid, H. Steerable wedge filters for local orientation analysis. *IEEE Trans. on Image Processing*, 5:1377–1382, 1996.

[Simoncelli und Freeman, 1995] Simoncelli, E.P., Freeman, W.T. The steerable pyramid: A flexible architecture for multiscale derivative computation. In *Proc. IEEE Int. Conference on Image Processing (ICIP)*, S. 444–447, 1995.

[Sirovich und Kirby, 1987] Sirovich, I., Kirby, M. Low-dimensional procedures for the characterization of human faces. *J. Optical Society of America A*, 4(3):519–524, 1987.

[Skodras et al., 2001] Skodras, A., Christopoulos, C., Ebrahimi, T. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001.

[Sloane und Conway, 1982] Sloane, N.J.A., Conway, J.H. Voronoi regions of lattices, second moments of polytopes, and quantization. *IEEE Trans. on Information Theory*, 28:211–226, 1982.

[Smith und Barnwell, 1986] Smith, M.J.T., Barnwell, T.P. Exact reconstruction techiniques for tree-structured subband coders. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 34:434–441, 1986.

[Sporring, 1997] Sporring, J. *Gaussian scale space theory*. Kluwer, Dordrecht, 1997.

[Stallings, 1972] Stallings, W.W. Recognition of printed Chinese characters by automatic pattern analysis. *Computer Graphics and Image Processing*, 1:47–65, 1972.

[Stearns, 1976] Stearns, S.D. On selecting features for pattern classifiers. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. 71–75. Coronado, California, USA, 1976.

[Stockman, 1978] Stockman, G. Defining and extracting waveform primitives for linguistic analysis. In *Proc. 4. Int. Conf. on Pattern Recognition*, S. 696–700. Kyoto, Japan, 1978.

[Story und Titze, 1998] Story, B.H., Titze, I.R. Parameterization of vocal tract area functions by empirical orthogonal modes. *J. Phonetics*, 26:223–260, 1998.

[Strube, 1985] Strube, H.W. A computationally efficient basilar membrane model. *Acustica*, 58:207–214, 1985.

[Suk und Flusser, 1996] Suk, T., Flusser, J. Vertex-based features for recognition of projectively deformed polygons. *Pattern Recognition*, 29:361–3677, 1996.

[Suk und Flusser, 2003] Suk, T., Flusser, J. Combined blur and affine moment invariants and their use in pattern recognition. *Pattern Recognition*, 36:2895–2907, 2003.

[Sung und Poggio, 1998] Sung, K.-K., Poggio, T. Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:39–50, 1998.

[Tan und Yan, 2001] Tan, T., Yan, H. Object recognition based on fractal neighbor distance. *Signal Processing*, 81:2105–2129, 2001.

[Teager, 1980] Teager, H.M. Some observations on oral air flow during phonation. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 28(5):599–601, 1980.

[Teague, 1980] Teague, M.R. Image analysis via the general theory of moments. *Journal Optical Society of America*, 70:920–930, 1980.

[Teh und Chin, 1988] Teh, C.-H., Chin, R. On image analysis by the method of moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10:496–512, 1988.

[Teuner et al., 1995] Teuner, A., Pichler, O., Hosticka, B.J. Unsupervised texture segmentation of

images using tuned matched Gabor filters. *IEEE Trans. on Image Processing*, 4:863–870, 1995.

[Therrien, 1975] Therrien, C.W. Eigenvalue properties of projection operators and their application to the subspace method of feature selection. *IEEE Trans. on Computers*, 24:944–948, 1975.

[Thomas, 1986] Thomas, T.J. A finite element model of fluid flow in the vocal tract. *Computer Speech & Language*, 1:131–151, 1986.

[Titsias und Likas, 2001] Titsias, M.K., Likas, A.C. Shared kernel models for class conditional density estimation. *IEEE Trans. on Neural Networks*, 12(5):987–997, 2001.

[Tjostheim und Sandvin, 1979] Tjostheim, D., Sandvin, O. Multivariate autoregressive feature extraction and the recognition of multichannel waveforms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:80–86, 1979.

[Tsai und Tsai, 2002] Tsai, D.M., Tsai, Y.H. Rotation invariant pattern matching with color ring projection. *Pattern Recognition*, 35:131–141, 2002.

[Turk und Pentland, 1991] Turk, M., Pentland, A. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.

[Tuzikov et al., 2003] Tuzikov, A.V., Sheynin, S.A., Vasiliev, P.V. Computation of volume and surface body moments. *Pattern Recognition*, 36:2521–2529, 2003.

[Unser, 1986a] Unser, M. Local linear transforms for texture measurements. *Signal Processing*, 11:61–79, 1986a.

[Unser, 1986b] Unser, M. Sum and difference histograms for texture analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:118–125, 1986b.

[Unser et al., 1991] Unser, M., Aldroubi, A., Eden, M. Fast B-spline transform for continuous image representation and interpolation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:277–285, 1991.

[Unser et al., 1993a] Unser, M., Aldroubi, A., Eden, M. B-spline signal processing: Part I – theory. *IEEE Trans. on Signal Processing*, 41:821–833, 1993a.

[Unser et al., 1993b] Unser, M., Aldroubi, A., Eden, M. B-spline signal processing: Part II – efficient design and applications. *IEEE Trans. on Signal Processing*, 41:834–848, 1993b.

[Unser und Blu, 2003a] Unser, M., Blu, T. Mathematical properties of the JPEG2000 wavelet filters. *IEEE Trans. on Image Processing*, 12:1080–1090, 2003a.

[Unser und Blu, 2003b] Unser, M., Blu, T. Wavelet theory demystified. *IEEE Trans. on Signal Processing*, 51:470–483, 2003b.

[Unser und Eden, 1989] Unser, M., Eden, M. Multiresolution feature extraction and selection for texture segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:717–728, 1989.

[Usevitch, 2001] Usevitch, B.E. A tutorial on modern lossy image compression: Foundations of JPEG 2000. *IEEE Signal Processing Magazine*, 18(5):22–35, 2001.

[Vajda, 1970] Vajda, I. Note on discrimination information and variation. *IEEE Trans. on Information Theory*, 16:771–773, 1970.

[van Gool et al., 1985] van Gool, L., Dewaele, P., Oosterlinck, A. Texture analysis anno 1983. *Computer Vision, Graphics, and Image Processing*, 29:336–357, 1985.

[van Gool et al., 1996] van Gool, L., Moons, T., Ungureanu, D. Affine/photometric invariants for planar intensity patterns. In *Proc. European Conference on Computer Vision (ECCV)*, S. 642–651. Springer, Lecture Notes in Computer Science Nr. 1064, 1996.

[van Otterloo und Young, 1978] van Otterloo, P.J., Young, I.T. A distribution-free geometric upper bound for the probability of error of a minimum distance classifier. *Pattern Recognition*, 10:281–286, 1978.

[Vapnik, 1995] Vapnik, V.N. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[Vetterli und Herley, 1992] Vetterli, M., Herley, C. Wavelets and filter banks: Theory and design. *IEEE Trans. on Signal Processing*, 40:2207–2232, 1992.

[Vidal et al., 2005] Vidal, R., Ma, Y., Sastri, S. Generalized principal component analysis (GPCA). *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1945–1959, 2005.

[Viikki und Laurila, 1997] Viikki, O., Laurila, K. Noise robust HMM-based speech recognition using segmental cepstral feature vector normalization. In *Proc. ESCA-NATO Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*, S. 107–110. Pont-a-Mousson, France, 1997.

[Vilmansen, 1973] Vilmansen, T.R. Feature evaluation with measures of probabilistic dependence. *IEEE Trans. on Computers*, 22:381–388, 1973.

[Vrabie et al., 2004] Vrabie, V.D., Mars, J.I., Lacoume, J.-L. Modified singular value decomposition by means of independent component analysis. *Signal Processing*, 84:645–652, 2004.

[Wagner, 1999] Wagner, T. Texture analysis. In B. Jähne, H. Haußecker, P. Geißler, Hg., *Handbook of Computer Vision and Applications*, Bd. 2, S. 275–308. Academic Press, New York, USA, 1999.

[Wakita, 1973] Wakita, H. Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveform. *IEEE Trans. on Audio Electroacoustics*, 21:417–427, 1973.

[Wang et al., 2003] Wang, Z., Lee, Y., Fiori, S., Leung, C.-S., Zhu, Y.-S. An improved sequential method for principal component analysis. *Pattern Recognition Letters*, 24:1409–1415, 2003.

[Watanabe, 1965] Watanabe, S. Karhunen-Loeve expansion and factor analysis. *Transactions 4. Prague Conference on Information Theory*, S. 635–660, 1965.

[Weiss, 1993] Weiss, I. Geometric invariants and object recognition. *Int. Journal of Computer Vision*, 10:207–231, 1993.

[Weng et al., 2003] Weng, J., Zhang, Y., Hwang, W.-S. Candid covariance-free incremantal principal component analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1034–1040, 2003.

[White und Neeley, 1976] White, G.M., Neeley, R.B. Speech recognition experiments with linear prediction, bandpass filtering, and dynamic programming. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 24:183–188, 1976.

[Whitney, 1971] Whitney, A.W. A direct method of nonparametric measurement selection. *IEEE Trans. on Computers*, 20:1100–1103, 1971.

[Wilkins, 1962/63] Wilkins, H.J. Householders method for symmetric matrices. *Numerische Mathematik*, 4:354–376, 1962/63.

[Wilks, 1962] Wilks, S. *Mathematical Statistics*. J. Wiley, New York, 1962.

[Winkler, 1977] Winkler, G. *Stochastische Systeme, Analyse and Synthese*. Akademische Verlagsgesellschaft, Wiesbaden, 1977.

[Wong und Hall, 1978] Wong, R.Y., Hall, E.L. Scene matching with invariant moments. *Computer Graphics and Image Processing*, 8:16–24, 1978.

[Wood und Treitel, 1975] Wood, L.C., Treitel, S. Seismic signal processing. *Proc. IEEE*, 63:649–661, 1975.

[Wu et al., 1991] Wu, C.S., Nguyen, V.V., Sabrin, H., Kushner, W., Damoulakis, J. Fast self-adapting broadband noise removal in the cepstral domain. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 957–960. Toronto, 1991.

[Wu und Chiu, 2001] Wu, J.-M., Chiu, S.-J. Independent component analysis using Potts model. *IEEE Trans. on Neural Networks*, 12:202–211, 2001.

[Xu et al., 2004] Xu, Y., Yang, J.-Y., Jin, Z. A novel method for Fisher discriminant analysis. *Pattern Recognition*, 37:381–384, 2004.

[Yang et al., 2004] Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y. Two-dimensional PCA: A new approach to appearance based face representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:131–137, 2004.

[Yang et al., 2005] Yang, J., Zhang, D., Yong, X., Yang, J.-Y. Two-dimensional discriminant transform for face recognition. *Pattern Recognition*, 38:1125–1129, 2005.

[Yang und Chen, 2002] Yang, J.-F., Chen, F.-K. Recursive discrete Fourier transform with unified IIR filter structures. *Signal Processing*, 82:31–41, 2002.

[Yang und Albregtsen, 1994] Yang, L., Albregtsen, F. Fast computation of invariant geometric mo-

ments: A new method giving correct results. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. Vol. I, 201–204. IEEE Computer Society, Jerusalem, 1994.

[Yang, 2002] Yang, M.H. Kernel eigenfaces versus kernel Fisher faces. In *Proc. 5th IEEE Int. Conference on Automatic Face and Gesture Recognition*, S. 215–220. Washington, D.D., 2002.

[Yap et al., 2003] Yap, P.-T., Paramesran, R., Ong, S.-H. Image analysis by Krawtchouk moments. *IEEE Trans. on Image Processing*, 12:1367–1377, 2003.

[Yapanel et al., 2001] Yapanel, U., Hansen, J.H.L., Sarikaya, R., Pellom, B. Robust digit recognition in noise: An evaluation using the AURORA corpus. In *Proc. European Conference on Speech Communication and Technology*, S. 209–212. Aalborg, Denmark, 2001.

[Ye und Li, 2004] Ye, J., Li, Q. LDA/QR: An efficient and effective dimension reduction algorithm and its theoretical foundation. *Pattern Recognition*, 37:851–854, 2004.

[Yilmaz und Gökmen, 2001] Yilmaz, A., Gökmen, M. Eigenhill vs. eigenface and eigenedge. *Pattern Recognition*, 34(1):181–184, 2001.

[Yost, 1994] Yost, W.A. *Fundamentals of Hearing: An Introduction*. Academic Press, San Diego, CA, USA, 1994.

[Yu, 2001] Yu, W. *Local Orientation Analysis in Images and Image Sequences Using Steerable Filters*. Berichte aus der Informatik. Shaker, Aachen, 2001.

[Zahn und Roskies, 1972] Zahn, C.T., Roskies, R.Z. Fourier descriptors for plane closed curves. *IEEE Trans. on Computers*, 21:269–281, 1972.

[Zhang und Tan, 2002] Zhang, J., Tan, T. Brief review of invariant texture analysis methods. *Pattern Recognition*, 35:735–747, 2002.

[Zhang und Benveniste, 1992] Zhang, Q., Benveniste, A. Wavelet networks. *IEEE Trans. on Neural Networks*, 3:889–898, 1992.

[Zhao et al., 1995] Zhao, Y., Applebaum, T., Hanson, B. Acoustic normalization and adaptation for microphone-channel characteristics. In *Proc. International Conference on Acoustics (ISBN 82-595-8995-8)*, Bd. 3, S. 189–192. Trondheim, Norway, 1995.

[Zhou et al., 2001] Zhou, G., Hansen, H.L., Kaiser, J.F. Nonlinear feature based classification of speech under stress. *IEEE Trans. on Speech and Audio Processing*, 9(3):201–216, 2001.

[Zhu et al., 2000] Zhu, S.C., Liu, X.W., Wu, Y.N. Exploring texture ensembles by efficient Markov chain Monte Carlo – toward a "trichromacy" theory of texture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:554–569, 2000.

[Zhu et al., 1997] Zhu, S.C., Wu, Y.N., Mumford, D.B. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9:1627–1660, 1997.

[Zhu et al., 1998] Zhu, S.C., Wu, Y.N., Mumford, D.B. Filters, random fields, and maximum entropy (FRAME): Towards a unified theory of texture modeling. *Int. Journal of Computer Vision*, 27(2):1–20, 1998.

[Zhu et al., 2002] Zhu, Y., De Silva, L.C., Ko, C.C. Using moment invariants and HMM in facial expression recognition. *Pattern Recognition Letters*, 23:83–91, 2002.

[Zwicker und Feldtkeller, 1967] Zwicker, E., Feldtkeller, R. *Das Ohr als Nachrichtenempfänger*. Hirzel Verlag, Stuttgart, Germany, 1967.

Kapitel 4

Numerische Klassifikation (VK.2.3.3, 07.09.2005)

Die in den vorangehenden beiden Kapiteln erörterten Verarbeitungsmethoden erlauben es, ein aufgenommenes Muster ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x})$ in einen Merkmalsvektor ${}^{\varrho}\boldsymbol{c}$ zu transformieren. Die grundlegende Voraussetzung ist, dass die erhaltenen Merkmale Postulat 3 aus Abschnitt 1.3 genügen. Es bleibt nun noch die Aufgabe, den Merkmalsvektor einer Klasse Ω_κ zuzuordnen, also die in (1.3.6), S. 21, angegebene Abbildung

$$ {}^{\varrho}\boldsymbol{c} \rightarrow \kappa \in \{1, ..., k\} \quad \text{oder} \quad {}^{\varrho}\boldsymbol{c} \rightarrow \kappa \in \{0, 1, ..., k\} $$

festzulegen und damit eine Klassifikation durchzuführen. Da die Komponenten ${}^{\varrho}c_\nu$ des Vektors ${}^{\varrho}\boldsymbol{c}$ gemäß (3.1.2), S. 163, reelle Zahlen sind, wird diese Abbildung als **numerische Klassifikation** bezeichnet. Es wird sich zeigen, dass zu ihrer Durchführung zum Teil umfangreiche numerische Rechnungen erforderlich sind. Die Klassifikation ist der letzte der in Bild 1.4.1, S. 26, angegebenen Verarbeitungsschritte, und damit ist die Klassifikationsaufgabe gelöst.

Da die Klassifikation von Merkmalsvektoren eine klar definierte abgegrenzte Aufgabe ist, gibt es dafür einige theoretisch wohl begründete Ansätze. In diesem Kapitel werden die folgenden Punkte behandelt:

1. Klassifikation mit minimalem Risiko – das Klassifikationsproblem wird als Optimierungsproblem im Rahmen der statistischen Entscheidungstheorie formuliert und gelöst.
2. Statistische Klassifikatoren – die von der statistischen Entscheidungstheorie geforderten statistischen Vorkenntnisse werden geschätzt.
3. Support Vektor Maschine – es werden die Merkmalsvektoren aus der Trainingsmenge, die das minimale empirische Risiko ergeben, zur Klassifikation verwendet.
4. Polynomklassifikator – das Klassifikationsproblem wird als Approximation einer idealen Trennfunktion formuliert und gelöst.
5. Neuronale Netze – eine vorgegebenen Trennfunktion wird mit gekoppelten künstlichen Neuronen approximiert.
6. Andere Klassifikatortypen – einige weitere Ansätze, darunter nichtlineare Normierung.
7. Klassifikation im Kontext – es wird eine optimale Folge von Einzelentscheidungen berechnet.
8. Unüberwachtes Lernen (Training) – Anpassung des Klassifikators an die zu klassifizierenden Muster durch Auswertung einer unklassifizierten Stichprobe.
9. Dimensionierungsprobleme – einige Beziehungen zwischen Fehlerwahrscheinlichkeit des Klassifikators, Umfang der Stichprobe und Zahl der Merkmale.

Zur Definition eines Klassifikators gibt es unterschiedliche Ansätze, von denen hier vier vorgestellt werden, nämlich der statistisch optimale Klassifikator in Abschnitt 4.1 und Definition 4.2, die Support Vektor Maschine in Abschnitt 4.3 und Definition 4.10, der Polynomklassifikator in Abschnitt 4.4 und Definition 4.13 sowie das Mehrschichtperzeptron in Abschnitt 4.5 mit dem Fehlermaß (4.5.13). Nach experimentellen Befunden sind alles sehr leistungsfähige Ansätze für die Klassifikation; ihre Einzelheiten gehen aus den genannten Abschnitten hervor.

Es wird daran erinnert, dass die Probleme der *Klassifikation* und *Regression* eng verwandt sind, wie bereits in Abschnitt 1.3 kurz erwähnt wurde. In beiden Fällen kommt es darauf an, aus einer Stichprobe mit Zusatzinformation y gemäß (1.3.1), S. 19, eine Funktion zu berechnen, mit der die Zusatzinformation für *nicht in der Stichprobe enthaltene* Muster möglichst zuverlässig geschätzt werden kann, d. h. die eine möglichst gute Generalisierung erlaubt. Bei der Klassifikation ist die zu schätzende Zusatzinformation die Musterklasse, also $y \in \{1, 2, \ldots, k\}$; bei der Regression ist sie ein Funktionswert, also $y \in \mathbb{R}$ oder allgemeiner $y \in \mathbb{R}^n$. Die Schätzfunktion hat i. Allg. die Form $y = d(\boldsymbol{f}, \boldsymbol{a})$, wobei $\boldsymbol{a}$ ein wählbarer Parametervektor ist. Zur Berechnung der Schätzfunktion gibt es im Wesentlichen zwei grundsätzliche Vorgehensweisen, von denen jede wieder zahlreiche Varianten aufweist.

1. Man ermittelt zunächst ein *stochastisches Modell* der Beobachtungen $\boldsymbol{f}$ in Form der klassenbedingten Verteilungsdichten der Muster $\boldsymbol{f}$ bzw. der daraus extrahierten Merkmale $\boldsymbol{c}$. Wenn die Verteilungsdichten bekannt sind, lassen sich Klassifiaktions– und Regressionsproblem lösen. Diese Vorgehensweise ist die Basis der statistischen Klassifikatoren.
2. Man bestimmt die Schätzfunktion *direkt*, insbesondere ohne vorher Verteilungsdichten von Beobachtungen zu bestimmen. Dieses ist die Vorgehensweise bei Support Vektor Maschinen, Polynomklassifikatoren und Mehrschichtperzeptrons.

Satz 4.14, S. 371, zeigt, dass es zwischen beiden Vorgehensweisen Beziehungen gibt. Daraus geht auch hervor, dass die allgemeine Lösung des Regressionsproblems ebenfalls statistische Information in Form von Verteilungsdichten (zur Berechnung des in Satz 4.14 auftretenden Erwartungswertes) erfordert. Zur Vereinfachung analytischer und numerischer Rechnungen wird allerdings oft statt der allgemeinen Schätzfunktion $y = d(\boldsymbol{f}, \boldsymbol{a})$ eine spezielle im Parametervektor $\boldsymbol{a}$ *lineare* Schätzfunktion $y = d(\boldsymbol{f}, \boldsymbol{a}) = \boldsymbol{a}^{\mathsf{T}} \boldsymbol{\varphi}(\boldsymbol{f})$ gewählt. In diesem Falle kann man die Schätzfunktion auch direkt aus der Stichprobe berechnen.

4.1 Statistische Entscheidungstheorie (VA.1.2.3, 13.04.2004)

4.1.1 Ansatz

Im folgenden wird die Klassifikation eines Musters als das Problem des Treffens einer *Entscheidung* aufgefasst, wobei diese entweder für eine der k Klassen $\Omega_\kappa\,, \kappa = 1, \ldots, k$ oder für die Rückweisungsklasse Ω_0 zu erfolgen hat. Die Entscheidung erfolgt auf der Basis einer *Beobachtung*, die in dem gegebenen Muster ${}^{\varrho}\boldsymbol{f}$ bzw. in dem daraus extrahierten Merkmalsvektor ${}^{\varrho}\boldsymbol{c}$ besteht. Ein automatisches System wird i. Allg. sehr viele Muster zu klassifizieren haben, sodass eine gute Systemleistung für möglichst viele bzw. für alle in Frage kommenden Muster erwünscht ist.

Wir gehen davon aus, dass jede Entscheidung individuelle *Kosten* verursacht, die zum einen vom Problemkreis abhängen, zum anderen davon, ob die Entscheidung richtig oder falsch war oder durch Rückweisung vermieden wurde. Die Kosten können zum einen als Vielfache einer Währungseinheit verstanden werden, zum anderen als *Gewichtsfaktoren*, mit denen richtigen (oder falschen) Entscheidungen bei bestimmten Klassen ein größeres Gewicht zugemessen wird als bei anderen Klassen. Wenn ein System viele Entscheidungen trifft, also viele Muster zu klassifizieren hat, sind die *mittleren Kosten* bei der Klassifikation eine mögliche Kenngröße für die Güte des Systems; die mittleren Kosten werden auch als das *Risiko* bezeichnet.

Es ist klar, dass ein System Entscheidungen nach unterschiedlichen Kriterien oder *Entscheidungsregeln* treffen kann. Damit ergibt sich zunächst unter Verzicht auf eine formale Darstellung der Standardansatz der *Entscheidungstheorie*, der auch in der Klassifikation von Mustern eine zentrale Bedeutung hat:

Definition 4.1 *Der* **optimale Klassifikator** *arbeitet so, dass die mittleren Kosten bzw. das Risiko bei der Klassifikation minimiert werden.*

Natürlich wird jeder gute Klassifikator versuchen, den optimalen Klassifikator zu approximieren. Allerdings hängt Optimalität vom verwendeten Optimierungskriterium ab. Der entscheidungstheoretische Ansatz geht, wie im folgenden gezeigt wird, von *vollständigen* statistischen Kenntnissen aus, die dann in einem als Training bezeichneten Prozess geschätzt bzw. approximiert werden. Es wird darauf hingewiesen, dass Optimalität auch anders definiert werden kann. Ein Beispiel dafür ist die Verwendung des *empirischen Risikos* in Definition 4.10, S. 363, wobei von Anfang an der Einfluss einer *endlichen Stichprobe* in den Optimierungsprozess einbezogen wird.

Der Weg zur formalen Berechnung des optimalen Klassifikators ist mit obiger Definition vorgezeichnet. Zunächst werden die mittleren Kosten beim Betrieb des Klassifikators berechnet. Diese hängen von einer noch frei wählbaren Entscheidungsregel ab. Danach wird die Entscheidungsregel berechnet, die die mittleren Kosten minimiert. Es wird sich zeigen, dass die mittleren Kosten berechnet werden können, wenn vollständige statistische Information über die Muster vorausgesetzt wird. Weiter wird sich zeigen, dass bei spezieller Wahl der Kosten für richtige und falsche Entscheidung die mittleren Kosten gleich der *Fehlerwahrscheinlichkeit* werden. Damit hat man einen leistungsfähigen und flexiblen Ansatz zum Treffen von Entscheidungen.

Die Ergebnisse dieses Abschnitts sind in Satz 4.1, S. 310, Satz 4.2, S. 314, sowie Satz 4.3, S. 315, zusammengefasst. Sie bilden zusammen mit den Verallgemeinerungen in Abschnitt 4.1.6 die Grundlage für die Klassifikation von Mustern auf der Basis des entscheidungstheoretischen Ansatzes.

Als *Vorteile* des entscheidungstheoretischen bzw. statistischen Ansatzes sind zu nennen:

- Das einfache Beispiel in (1.5.1), S. 28, zeigt, dass die Wahrscheinlichkeitstheorie ein theoretisch fundierter Ansatz ist, mehrere unsichere Einzelbeobachtungen (Merkmale) so zu kombinieren, dass das Gesamtergebnis (die Klassifikation) immer sicherer wird.
- Es gibt zur Ermittlung von Wahrscheinlichkeiten bzw. von Parametern von Wahrscheinlichkeitsdichten aus Beobachtungen, nämlich Stichproben ω in (1.3.1), eine fundierte Schätztheorie, insbesondere die in Abschnitt 4.2.2 eingeführten Maximum-likelihood- und BAYES-Schätzwerte.
- Man kann sehr allgemeine statistische Modelle angeben, die auf einheitlicher theoretischer Basis effiziente Lösungen für die Schätzung der Modellparameter und für die Klassifikation von Mustern, im Falle von zwei– oder dreidimensionalen Objekten auch für deren Lokalisation, erlauben. Beispiele dafür enthält Abschnitt 4.2.1 für die Klassifikation von Merkmalsvektoren und Abschnitt 4.7 für die Klassifikation im Kontext.

Als *Nachteile* des statistischen Ansatzes sind zu nennen:

- Die Ermittlung von statistischen Modellen, die den Problemkreis hinreichend genau charakterisieren, ist i. Allg. ein schwieriges Problem.
- Es gibt eine Reihe von leistungsfähigen Alternativen zu statistischen Klassifikatoren, wie in einigen Abschnitten dieses Kapitels noch gezeigt wird.

4.1.2 Voraussetzungen

Die wesentliche Voraussetzung bei der Durchrechnung des obigen Ansatzes besteht darin, dass vollständige Kenntnisse über die statistischen Eigenschaften der Muster aus einer Klasse Ω_κ gegeben sind. Es wird vorausgesetzt, dass eine n–dimensionale parametrische Familie $\widetilde{p}(\boldsymbol{c}\,|\,\boldsymbol{a})$ von Verteilungsdichten bekannt ist und dass die **klassenbedingten Verteilungsdichten** $p(\boldsymbol{c}\,|\,\Omega_\kappa)$ der Merkmalsvektoren Elemente dieser Familie sind, wobei $\boldsymbol{a}$ ein möglicherweise unbekannter Parametervektor ist. Damit gilt

$$p(\boldsymbol{c}\,|\,\Omega_\kappa) = p(\boldsymbol{c}\,|\,\boldsymbol{a}_\kappa) \in \widetilde{p}(\boldsymbol{c}\,|\,\boldsymbol{a}) = \{p(\boldsymbol{c}\,|\,\boldsymbol{a})\,|\,\boldsymbol{a} \in \mathbb{R}_{\boldsymbol{a}}\}\ . \tag{4.1.1}$$

Mit (4.1.1) wird zum Ausdruck gebracht, dass a priori Information über die Klassen in Form der parametrischen Familie $\widetilde{p}$ vorhanden ist. Die Bestimmung der Dichte $p(\boldsymbol{c}\,|\,\Omega_\kappa)$ reduziert sich damit auf die Bestimmung der unbekannten Parameter $\boldsymbol{a}_\kappa$.

In einem bestimmten Problemkreis Ω treten Muster aus der Klasse Ω_κ mit einer bestimmten **a priori Wahrscheinlichkeit** $p(\Omega_\kappa) = p_\kappa$ auf, von der vorausgesetzt wird, dass sie ebenfalls bekannt ist. Aufgrund der Definition bedingter Dichten ergibt sich die **Verbunddichte** zu

$$p(\boldsymbol{c}, \Omega_\kappa) = p_\kappa p(\boldsymbol{c}\,|\,\Omega_\kappa) = p(\boldsymbol{c})p(\Omega_\kappa\,|\,\boldsymbol{c})\ . \tag{4.1.2}$$

Daraus ergibt sich die wichtige BAYES-**Regel**

$$\boxed{p(\Omega_\kappa\,|\,\boldsymbol{c}) = \frac{p_\kappa p(\boldsymbol{c}\,|\,\Omega_\kappa)}{p(\boldsymbol{c})}\ .} \tag{4.1.3}$$

Beobachtete Muster lassen sich als Ergebnis eines Zufallsprozesses auffassen. Innerhalb des Problemkreises Ω wird zufällig eine Klasse ausgewählt, wobei die Klasse Ω_κ mit der Wahrscheinlichkeit p_κ gewählt wird. Nach Wahl von Ω_κ wird eine Beobachtung ${}^{\varrho}\boldsymbol{c}$ – nämlich der Merkmalsvektor eines Musters – der Zufallsvariablen $\boldsymbol{c}$ gemacht, wobei $\boldsymbol{c}$ die bedingte Dichte

$p(\boldsymbol{c}|\Omega_\kappa)$ hat. Ergebnis des Zufallsprozesses ist also das Paar $(\kappa, \boldsymbol{c})$, d. h. eine Klassennummer (Zusatzinformation, s. (1.3.1), S. 19) und eine Beobachtung (Wert einer Zufallsvariablen). Die BAYES-Regel ist die Basis der statistischen Inferenz, mit der, wie schon in Abschnitt 1.5 erwähnt, die a priori Wahrscheinlichkeit p_κ einer Klasse *vor* Beobachtung eines Merkmalsvektors $\boldsymbol{c}$ in die a posteriori Wahrscheinlichkeit $p(\Omega_\kappa|\boldsymbol{c})$ *nach* Beobachtung eines Merkmalsvektors transformiert wird. Sie bildet eine mathematisch einwandfreie Basis für statistische Inferenzen.

Wenn Muster klassifiziert werden, können Fehlklassifikationen auftreten. Um den Einfluss von Fehlern bei der Beurteilung der Klassifikatorleistung zu erfassen, werden den Fehlern wie erwähnt **Kosten** zugeordnet. Mit

$$r_{\lambda\kappa} = r(\Omega_\lambda|\Omega_\kappa)\ , \quad \lambda = 0, 1, \ldots, k\ , \quad \kappa = 1, \ldots, k \tag{4.1.4}$$

werden die Kosten bezeichnet, die entstehen, wenn man ein Muster nach Ω_λ klassifiziert, obwohl es tatsächlich aus Ω_κ stammt. Es wird vorausgesetzt, dass die Kosten $r_{\lambda\kappa}$ bekannt sind und der Bedingung

$$0 \leq r_{\kappa\kappa} < r_{0\kappa} < r_{\lambda\kappa}\ , \quad \lambda \neq \kappa \tag{4.1.5}$$

genügen, d. h. Kosten sind nicht negativ und die Kosten einer richtigen Entscheidung sind geringer als die einer Rückweisung, und diese sind wiederum geringer als die Kosten einer Fehlklassifikation. Bei geeigneter Wahl der Kosten ergeben sich auch Beziehungen zur Fehlerwahrscheinlichkeit, wie später noch gezeigt wird.

Die frei wählbare **Entscheidungsregel** wird mit $\delta(\Omega_\lambda|\boldsymbol{c})$ bezeichnet und gibt die *Wahrscheinlichkeit* an, mit der man sich für die Klasse Ω_λ entscheidet, wenn der Merkmalsvektor $\boldsymbol{c}$ beobachtet wurde. Eine solche **randomisierte Entscheidungsregel** wird wegen ihrer größeren Allgemeinheit hier zunächst zugelassen. Allerdings wird sich zeigen, dass der optimale Klassifikator eine *nicht* randomisierte Regel verwendet. Diese ist ein Spezialfall der randomisierten Regel, bei dem für jeden Merkmalsvektor mit der Wahrscheinlichkeit 1 eine Entscheidung für genau eine der möglichen Klassen erfolgt und alle anderen die Wahrscheinlichkeit 0 haben. In der Literatur wird vielfach der optimale Klassifikator bestimmt, wenn von vornherein nur eine nicht randomisierte Regel zugelassen wird. Es wird noch vorausgesetzt, dass

$$\sum_{\lambda=1}^{k} \delta(\Omega_\lambda|\boldsymbol{c}) = 1 \quad \text{oder} \quad \sum_{\lambda=0}^{k} \delta(\Omega_\lambda|\boldsymbol{c}) = 1 \tag{4.1.6}$$

ist, d. h. auch bei der randomisierten Regel erfolgt immer eine Entscheidung für eine der vorhandenen Klassen. Der wesentliche Unterschied zwischen nicht randomisierter und randomisierter Regel ist, dass bei mehrfacher Beobachtung des *gleichen* Merkmalsvektors von ersterer auch stets für die *gleiche* Klasse entschieden wird, von letzterer dagegen *nicht*. Intuitiv scheint eine randomisierte Regel für die Mustererkennung wenig sinnvoll, jedoch ist zunächst offen, ob sich durch ihre Anwendung nicht ein geringeres Risiko ergibt.

Die wesentlichen Elemente eines Klassifikators sind also:

- Die a priori Wahrscheinlichkeiten p_κ der Klassen. Sie sind durch den Problemkreis bestimmt und müssen gegebenenfalls vom Entwickler eines Klassifikators geschätzt werden.
- Die bedingten Verteilungsdichten $p(\boldsymbol{c}|\Omega_\kappa)$ der Merkmalsvektoren. Sie liegen durch den Problemkreis, die verwendeten Sensoren und die gewählte Vorverarbeitung und Merkmalsgewinnung fest und müssen vom Entwickler eines Klassifikators geschätzt werden.

- Die Kosten $r_{\lambda\kappa}$ von Entscheidungen. Ihre Wahl ist Sache des Entwicklers oder des Betreibers eines Klassifikators.
- Die Entscheidungsregel $\delta(\Omega_\kappa|\boldsymbol{c})$. Sie kann *frei gewählt* werden.

Damit ist es möglich, den optimalen Klassifikator anzugeben, der die mittleren Kosten minimiert, wie in Abschnitt 4.1.3 gezeigt wird. Ein wesentliches Problem ist natürlich die Ermittlung der geforderten Größen $p(\boldsymbol{c}|\Omega_\kappa)$, p_κ, $r_{\lambda\kappa}$. Auf die Bestimmung der statistischen Größen wird in Abschnitt 4.2.1 eingegangen.

Zur Ableitung der grundsätzlichen Ergebnisse gehen wir hier davon aus, das *alle erforderliche Information gegeben* ist, d. h. $p(\boldsymbol{c}|\Omega_\kappa), p_\kappa, r_{\lambda\kappa}$ seien *bekannt*.

Die Probleme der Realisierung statistischer Klassifikatoren resultieren vor allem daraus, dass die bedingten Dichten $p(\boldsymbol{c}|\Omega_\kappa)$ i. Allg. *unbekannt* sind und oft nur unvollkommen approximiert bzw. geschätzt werden können.

4.1.3 Die optimale Entscheidungsregel

Gesucht ist unter den obigen Voraussetzungen eine Entscheidungsregel, mit der man beobachtete Merkmalsvektoren $\boldsymbol{c}$ optimal im Sinne der Definition 4.1 klassifizieren kann. Dazu wird in den folgenden drei Schritten vorgegangen:

1. Berechnung der Verwechslungswahrscheinlichkeit $p(\Omega_\lambda|\Omega_\kappa)$ von Mustern in (4.1.8).
2. Berechnung des Risikos $V(\delta)$ in (4.1.10).
3. Festlegung der optimalen Entscheidungsregel δ^* in (4.1.16).

Um das Risiko oder die mittleren Kosten in Abhängigkeit von der verwendeten Entscheidungsregel zu berechnen, wird zunächst die Wahrscheinlichkeit $p(\Omega_\lambda, \boldsymbol{c}|\Omega_\kappa)$ berechnet, mit der ein Vektor $\boldsymbol{c}$ auftritt und nach Ω_λ klassifiziert wird, obwohl er aus Ω_κ stammt. Aufgrund der Definition bedingter Wahrscheinlichkeiten ist

$$\begin{aligned} p(\Omega_\lambda, \boldsymbol{c}|\Omega_\kappa) &= \frac{p(\Omega_\lambda, \Omega_\kappa, \boldsymbol{c})}{p(\Omega_\kappa)} \frac{p(\Omega_\kappa, \boldsymbol{c})}{p(\Omega_\kappa, \boldsymbol{c})} \\ &= p(\Omega_\lambda|\Omega_\kappa, \boldsymbol{c}) p(\boldsymbol{c}|\Omega_\kappa) \\ &= \delta(\Omega_\lambda|\Omega_\kappa, \boldsymbol{c}) p(\boldsymbol{c}|\Omega_\kappa) \\ &= \delta(\Omega_\lambda|\boldsymbol{c}) p(\boldsymbol{c}|\Omega_\kappa) \ . \end{aligned} \tag{4.1.7}$$

Die letzte Zeile von (4.1.7) ergibt sich daraus, dass die Wahrscheinlichkeit der Entscheidung für Ω_κ, wenn $\boldsymbol{c}$ beobachtet wurde und aus Ω_κ stammt, natürlich von Ω_κ unabhängig sein muss, da die richtige Klasse *nicht* mit beobachtet wird. Mit (4.1.7) erhält man die **Verwechslungswahrscheinlichkeit**, dass Muster aus Ω_κ nach Ω_λ eingeordnet werden, zu

$$p(\Omega_\lambda|\Omega_\kappa) = \int_{\mathbb{R}_{\boldsymbol{c}}} p(\Omega_\lambda, \boldsymbol{c}|\Omega_\kappa) \, \mathrm{d}\boldsymbol{c} = \int p(\boldsymbol{c}|\Omega_\kappa) \delta(\Omega_\lambda|\boldsymbol{c}) \, \mathrm{d}\boldsymbol{c} \ . \tag{4.1.8}$$

Wenn ein Muster aus Ω_κ nach Ω_λ klassifiziert wird, entstehen Kosten $r_{\lambda\kappa}$. Bei Anwendung der Entscheidungsregel δ tritt dieses Ereignis mit der durch (4.1.8) gegebenen Wahrscheinlichkeit $p(\Omega_\lambda|\Omega_\kappa)$ auf. Die durch Ω_κ bedingten mittleren Kosten oder der klassenbedingte mittlere Verlust ist dann aufgrund der Definition eines Erwartungswertes

$$V(\delta|\Omega_\kappa) = \sum_{\lambda=0}^{k} p(\Omega_\lambda|\Omega_\kappa) r_{\lambda\kappa}$$

$$= \sum_{\lambda} r_{\lambda\kappa} \int p(\boldsymbol{c}|\Omega_\kappa)\delta(\Omega_\lambda|\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\ . \tag{4.1.9}$$

Die mittleren Kosten oder das **Risiko** $V(\delta)$ bei der Klassifikation von Mustern aus dem Problemkreis Ω unter Verwendung der Entscheidungsregel δ erhält man durch Mittelung der bedingten Kosten über alle Klassen zu

$$\boxed{\begin{aligned} V(\delta) &= \sum_{\kappa=1}^{k} p_\kappa V(\delta|\Omega_\kappa) \\ &= \sum_{\kappa=1}^{k} p_\kappa \sum_{\lambda=0}^{k} r_{\lambda\kappa} \int_{\mathbb{R}_{\boldsymbol{c}}} p(\boldsymbol{c}|\Omega_\kappa)\delta(\Omega_\lambda|\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\ . \end{aligned}} \tag{4.1.10}$$

Die Summe in (4.1.9) erfolgt von 0 bis k, da dieses die von der Entscheidungsregel wählbaren Klassen sind, wenn man die Rückweisungsklasse Ω_0 als mögliche Entscheidung zulässt. Die Summe in (4.1.10) erfolgt von 1 bis k, da nur diese k Klassen auftreten können. Die $(k+1)$-te Rückweisungsklasse Ω_0 wurde ja in Abschnitt 1.2 nur eingeführt, um nicht sicher klassifizierbare Muster aus irgendeiner der k Klassen abweisen zu können.

Definition 4.2 *Der* **optimale Klassifikator** *wendet die Entscheidungsregel δ^* an, die definiert ist durch*

$$V(\delta^*) = \min_{\{\delta\}} V(\delta)\ , \quad \text{oder} \quad \delta^* = \operatorname*{argmin}_{\{\delta\}} V(\delta)\ . \tag{4.1.11}$$

Der beste Klassifikator ist also derjenige, der das Risiko bei der Klassifikation minimiert. Es wurde oben schon darauf hingewiesen, dass die Bezeichnung „optimaler Klassifikator“ natürlich relativ zu dem gewählten Gütekriterium – hier dem Risiko – zu verstehen ist. Wenn man außer dem Risiko noch andere Größen berücksichtigt, wie beispielsweise den erforderlichen Rechenaufwand, den Speicherplatz, die Rechengenauigkeit oder den Einfluss einer endlichen Stichprobe, dann kann natürlich ein anderer Klassifikator optimal im Sinne des neuen Kriteriums sein.

Es bleibt nun noch die Minimierung des Risikos $V(\delta)$, bzw. die Bestimmung der besten Entscheidungsregel δ^* gemäß (4.1.11). Dazu wird das Risiko in der Form

$$V(\delta) = \int_{\mathbb{R}_{\boldsymbol{c}}} \sum_{\lambda=0}^{k} \left[\sum_{\kappa=1}^{k} r_{\lambda\kappa} p_\kappa p(\boldsymbol{c}|\Omega_\kappa) \right] \delta(\Omega_\lambda|\boldsymbol{c})\,\mathrm{d}\boldsymbol{c} \tag{4.1.12}$$

geschrieben. Zur Abkürzung wird eine **Prüfgröße**

$$\boxed{u_\lambda(\boldsymbol{c}) = \sum_{\kappa=1}^{k} r_{\lambda\kappa} p_\kappa p(\boldsymbol{c}|\Omega_\kappa)\ , \quad \lambda = 0, 1..., k} \tag{4.1.13}$$

definiert. Sie enthält nur bekannte bzw. zunächst als bekannt vorausgesetzte Größen. Die Minimierung des Risikos ergibt sich aus der Überlegung, dass der Wert des Integrals dann ein Minimum annimmt, wenn für jeden Wert $\boldsymbol{c} \in \mathbb{R}_{\boldsymbol{c}}$ der Wert des Integranden minimiert wird. Für den Integranden in (4.1.12) gilt mit (4.1.6) die Abschätzung

$$I = \sum_{\lambda=0}^{k} u_\lambda(\boldsymbol{c})\delta(\Omega_\lambda|\boldsymbol{c}) \geq \sum_{\lambda=0}^{k} u_{\min}(\boldsymbol{c})\,\delta(\Omega_\lambda|\boldsymbol{c}) = u_{\min}(\boldsymbol{c})\ , \tag{4.1.14}$$

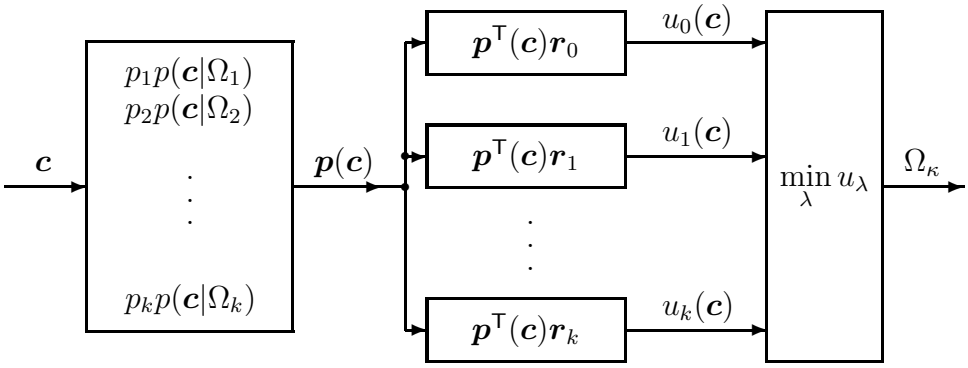


Bild 4.1.1: Die Struktur des optimalen Klassifikators, der das Risiko minimiert, gemäß (4.1.13), (4.1.16)

d. h. der Integrand kann als kleinsten Wert nur

$$u_{\min}(\boldsymbol{c}) = \min_\lambda u_\lambda(\boldsymbol{c}) \tag{4.1.15}$$

annehmen. Offensichtlich lässt sich für jeden Wert $\boldsymbol{c} \in \mathbb{R}_{\boldsymbol{c}}$ erreichen, dass der Integrand diesen kleinstmöglichen Wert annimmt und dadurch das Risiko minimiert wird – man muss nämlich nur die Entscheidungsregel geeignet wählen. Definiert man als **optimale Entscheidungsregel**

$$\boxed{\delta^*(\Omega_\kappa|\boldsymbol{c}) = \begin{cases} 1 & : \quad u_\kappa(\boldsymbol{c}) = \min\limits_\lambda u_\lambda(\boldsymbol{c}) \\ 0 & : \quad \lambda \neq \kappa\,, \quad \lambda = 0,1,\ldots,k\,, \end{cases}} \tag{4.1.16}$$

so nimmt der Integrand stets seinen minimalen Wert $u_{\min}$ an. Sollte es in (4.1.16) mehrere Minima geben, wird unter diesen ein beliebiges ausgewählt. Obwohl anfänglich eine randomisierte Entscheidungsregel zugelassen wurde, führt die Minimierung des Risikos auf eine nicht randomisierte Regel. Bei Beobachtung eines Merkmalsvektors $\boldsymbol{c}$ wählt also der optimale Klassifikator mit der Wahrscheinlichkeit 1 eine bestimmte Klasse aus. Mit den Vektoren

$$\begin{aligned} \boldsymbol{p}(\boldsymbol{c}) &= (p_1 p(\boldsymbol{c}|\Omega_1), \ldots, p_k p(\boldsymbol{c}|\Omega_k))^\mathsf{T} \\ \boldsymbol{r}_\lambda &= (r_{\lambda 1}, \ldots, r_{\lambda k})^\mathsf{T}\,, \quad \lambda = 0,1,\ldots,k \end{aligned} \tag{4.1.17}$$

lassen sich die Prüfgrößen in (4.1.13) auch kompakt als Skalarprodukt

$$u_\lambda(\boldsymbol{c}) = \boldsymbol{r}_\lambda^\mathsf{T} \boldsymbol{p}(\boldsymbol{c}), \quad \lambda = 0,1,\ldots,k \tag{4.1.18}$$

angeben. Die Struktur des Klassifikators zeigt Bild 4.1.1. Dieses Ergebnis ist im folgenden Satz zusammengefasst:

Satz 4.1 *Der optimale Klassifikator, der das Risiko $V(\delta)$ in (4.1.12) bei der Klassifikation minimiert, berechnet die $(k+1)$ Prüfgrößen $u_\lambda(\boldsymbol{c})$ gemäß (4.1.13). Er entscheidet sich stets für die Klasse Ω_κ, deren Prüfgröße u_κ den kleinsten Wert hat.*

4.1.4 Zwei spezielle Kostenfunktionen

In diesem Abschnitt werden zwei Klassifikatoren betrachtet, die sich aufgrund spezieller Kostenfunktionen ergeben.

Die (r_c, r_z, r_f)–Kostenfunktion

Als erstes wird die Kostenfunktion

$$\begin{aligned} r_{\kappa\kappa} &= r_c\ , \\ r_{0\kappa} &= r_z\ , \qquad\qquad \kappa, \lambda = 1, \ldots, k \\ r_{\lambda\kappa} &= r_f\ , \qquad \lambda \neq \kappa \end{aligned} \tag{4.1.19}$$

betrachtet. Es wird also angenommen, dass die Kosten der richtigen Klassifikation, bzw. der Rückweisung, bzw. der falschen Klassifikation für unterschiedliche Klassen jeweils gleich sind. Diese drei Möglichkeiten treten mit den Wahrscheinlichkeiten p_c, bzw. p_z, bzw. p_f auf. Aus der Definition des Risikos als mittlere Kosten ergibt sich

$$V(\delta) = r_c p_c + r_z p_z + r_f p_f\ . \tag{4.1.20}$$

Der optimale Klassifikator im Sinne von (4.1.11) berechnet wieder die $(k+1)$ Prüfgrößen (4.1.13). Aufgrund der spezialisierten Kostenfunktion ergibt sich die spezialisierte Entscheidungsregel in (4.1.25), sowie ein Zusammenhang zwischen Kosten, Fehler– und Rückweisungswahrscheinlichkeiten. Die Ergebnisse der folgenden längeren Rechnung sind in Satz 4.2 zusammengefasst.

Für die Kostenfunktion (4.1.19) ergeben sich die Prüfgrößen zu

$$\begin{aligned} u_0(\boldsymbol{c}) &= r_z \sum_{j=1}^{k} p_j p(\boldsymbol{c}|\Omega_j) \\ u_\lambda(\boldsymbol{c}) &= r_f \sum_{\substack{j=1 \\ j\neq\lambda}}^{k} p_j p(\boldsymbol{c}|\Omega_j) + r_c p_\lambda p(\boldsymbol{c}|\Omega_\lambda)\ . \end{aligned} \tag{4.1.21}$$

Eine Rückweisung erfolgt gemäß (4.1.16), wenn

$$u_0(\boldsymbol{c}) < u_\lambda(\boldsymbol{c})\ , \quad \lambda = 1, \ldots, k$$

ist. Das ergibt die Bedingung

$$\begin{aligned} r_z \sum_{j} p_j p(\boldsymbol{c}|\Omega_j) &< r_f \sum_{j\neq\lambda} p_j p(\boldsymbol{c}|\Omega_j) + r_c p_\lambda p(\boldsymbol{c}|\Omega_\lambda)\ , \\ r_z \sum_{j\neq\lambda} p_j p(\boldsymbol{c}|\Omega_j) + r_z p_\lambda p(\boldsymbol{c}|\Omega_\lambda) &< r_f \sum_{j\neq\lambda} p_j p(\boldsymbol{c}|\Omega_j) + r_c p_\lambda p(\boldsymbol{c}|\Omega_\lambda)\ , \\ p_\lambda p(\boldsymbol{c}|\Omega_\lambda) &< \frac{r_f - r_z}{r_z - r_c} \sum_{\substack{j=1 \\ j\neq\lambda}}^{k} p_j p(\boldsymbol{c}|\Omega_j)\ . \end{aligned} \tag{4.1.22}$$

Da man die Prüfgrößen $u_\lambda, \lambda \neq 0$ auch in der Form

$$u_\lambda(\boldsymbol{c}) = r_\mathrm{f} \sum_{j=1}^{k} p_j p(\boldsymbol{c}|\Omega_j) + (r_\mathrm{c} - r_\mathrm{f}) p_\lambda p(\boldsymbol{c}|\Omega_\lambda)$$

angeben kann, ist die Bedingung (4.1.22) äquivalent der Bedingung

$$\begin{aligned} p_\lambda p(\boldsymbol{c}|\Omega_\lambda) &< \alpha(\boldsymbol{c}) \,, \\ \alpha(\boldsymbol{c}) &= \frac{r_\mathrm{f} - r_\mathrm{z}}{r_\mathrm{f} - r_\mathrm{c}} \sum_{j=1}^{k} p_j p(\boldsymbol{c}|\Omega_j) \,, \end{aligned} \tag{4.1.23}$$

die für $\lambda = 1, \ldots, k$ gelten muss. Die letzte Form ist für die Berechnung vorzuziehen, da die rechte Seite von (4.1.23) eine von λ unabhängige Größe $\alpha(\boldsymbol{c})$ ist, die rechte Seite von (4.1.22) dagegen nicht.

Ist (4.1.23) oder (4.1.22) nicht für $\lambda = 1, \ldots, k$ erfüllt, so erfolgt eine Entscheidung für eine Klasse Ω_κ aus den k Klassen $\Omega_\lambda, \lambda = 1, \ldots, k$. Für die Prüfgröße $u_\kappa(\boldsymbol{c})$ dieser Klasse Ω_κ gilt wegen (4.1.16)

$$\begin{aligned} u_\kappa(\boldsymbol{c}) &< u_\lambda(\boldsymbol{c}) \,, \quad \lambda = 1, \ldots, k \,, \quad \lambda \neq \kappa \\ r_\mathrm{f} \sum_{j=1}^{k} p_j p(\boldsymbol{c}|\Omega_j) &+ (r_\mathrm{c} - r_\mathrm{f}) p_\kappa p(\boldsymbol{c}|\Omega_\kappa) < r_\mathrm{f} \sum_{j=1}^{k} p_j p(\boldsymbol{c}|\Omega_j) + (r_\mathrm{c} - r_\mathrm{f}) p_\lambda p(\boldsymbol{c}|\Omega_\lambda) \\ p_\kappa p(\boldsymbol{c}|\Omega_\kappa) &> p_\lambda p(\boldsymbol{c}|\Omega_\lambda) \,, \quad \lambda = 1, \ldots, k \,, \quad \lambda \neq \kappa \,. \end{aligned} \tag{4.1.24}$$

Offensichtlich braucht (4.1.23) nur für diesen Index κ geprüft zu werden.

Die allgemeine Entscheidungsregel (4.1.16) erhält also für die Kostenfunktion (4.1.19) wegen (4.1.23) und (4.1.24) die spezielle Form

$$\boxed{\begin{aligned} \delta^*(\Omega_\kappa|\boldsymbol{c}) &= 1 \,, \quad \text{wenn } p_\kappa p(\boldsymbol{c}|\Omega_\kappa) = \max_{\lambda \in \{1,\ldots,k\}} p_\lambda p(\boldsymbol{c}|\Omega_\lambda) \\ &\qquad\qquad \text{und} \quad p_\kappa p(\boldsymbol{c}|\Omega_\kappa) \geq \alpha(\boldsymbol{c}) \,, \\ \delta^*(\Omega_0|\boldsymbol{c}) &= 1 \,, \quad \text{wenn } p_\kappa p(\boldsymbol{c}|\Omega_\kappa) < \alpha(\boldsymbol{c}) \,. \end{aligned}} \tag{4.1.25}$$

Es wird nun gezeigt, dass der Klassifikator, der gemäß (4.1.25) das Risiko minimiert, äquivalent einem Klassifikator ist, der bei fest vorgegebener Rückweisungswahrscheinlichkeit $p_\mathrm{z} = p_{\mathrm{z}_0}$ die Fehlerwahrscheinlichkeit minimiert. Dafür wird zunächst die Rückweisungswahrscheinlichkeit p_z berechnet. Mit (4.1.8) und (4.1.23) gilt

$$\begin{aligned} p_\mathrm{z} &= p(\Omega_0) = \sum_{\kappa=1}^{k} p_\kappa p(\Omega_0|\Omega_\kappa) \\ &= \sum p_\kappa \int_{\mathbb{R}_{\boldsymbol{c}}} p(\boldsymbol{c}|\Omega_\kappa) \delta(\Omega_0|\boldsymbol{c}) \, \mathrm{d}\boldsymbol{c} \end{aligned}$$

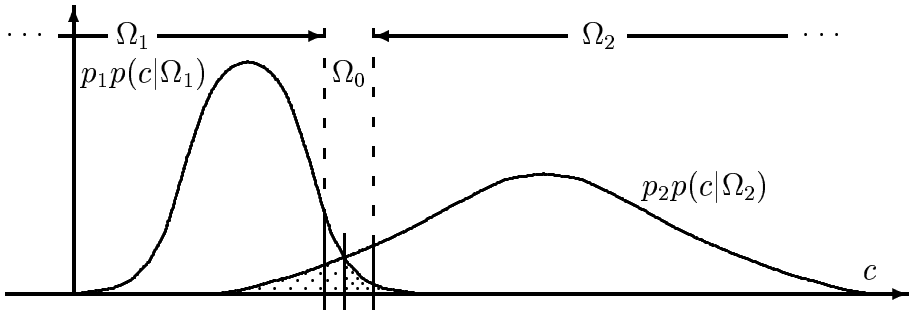


Bild 4.1.2: Zur Veranschaulichung von (4.1.25): Für $\beta = 0$ gibt es *keine* Zurückweisungen, dafür aber Fehlklassifikationen im punktierten Bereich; für $\beta = 0,74$ werden Muster im Bereich Ω_0 zurückgewiesen, die Zahl der Fehlklassifikationen sinkt auf Kosten der Rückweisungen und auch die Zahl der richtigen Entscheidungen nimmt ab

$$= \sum p_\kappa \int\limits_{\{\boldsymbol{c}\,|\,p_\lambda p(\boldsymbol{c}|\Omega_\lambda)<\alpha(\boldsymbol{c}),\ \lambda=1,\ldots k\}} p(\boldsymbol{c}|\Omega_\kappa)\,\mathrm{d}\boldsymbol{c} \tag{4.1.26}$$

Das Integral ist über den Bereich des Merkmalsraumes $\mathbb{R}_{\boldsymbol{c}}$ zu erstrecken, in dem Rückweisungen erfolgen. Für den optimalen Klassifikator ist dieser Bereich durch Bedingung (4.1.23) definiert. Durch Wahl der von den Kosten abhängigen Größe

$$\beta = \frac{r_\mathrm{f} - r_\mathrm{z}}{r_\mathrm{f} - r_\mathrm{c}}\,, \tag{4.1.27}$$

die in (4.1.23) und damit auch in (4.1.26) auftritt, lässt sich eine bestimmte Rückweisungswahrscheinlichkeit $p_\mathrm{z} = p_{\mathrm{z}_0}$ einstellen. Der Zusammenhang zwischen p_z und β wird sich zwar nur näherungsweise numerisch bestimmen lassen, jedoch ist dieses für das grundsätzliche Ergebnis belanglos; dieses besagt, dass das Rückweisungskriterium (4.1.23) äquivalent der Einstellung einer bestimmten Rückweisungswahrscheinlichkeit ist. Dieses wird in Bild 4.1.2 für den eindimensionalen Fall und zwei Klassen verdeutlicht.

Zwischen den Wahrscheinlichkeiten p_z, p_c, p_f besteht die Beziehung

$$p_\mathrm{z} + p_\mathrm{c} + p_\mathrm{f} = 1\,, \tag{4.1.28}$$

sodass die Minimierung von p_f bei festem $p_\mathrm{z} = p_{\mathrm{z}_0}$ der Maximierung von p_c entspricht. Die Wahrscheinlichkeit einer korrekten Entscheidung ergibt sich aus (4.1.8) unter Berücksichtigung der Rückweisungsbedingung zu

$$\begin{aligned} p_\mathrm{c} &= \sum_{\kappa=1}^{k} p_\kappa p(\Omega_\kappa|\Omega_\kappa) \\ &= \sum_{\kappa=1}^{k} p_\kappa \int\limits_{\{\boldsymbol{c}\,|\,p_\lambda p(\boldsymbol{c}|\Omega_\lambda)\geq\alpha(\boldsymbol{c})\}\ \text{für ein}\ \lambda\in\{1,\ldots,k\}} p(\boldsymbol{c}|\Omega_\kappa)\,\mathrm{d}\boldsymbol{c}\,. \end{aligned} \tag{4.1.29}$$

Die Entscheidungsregel δ ist so zu wählen, dass p_c maximiert wird, wenn man über den angegebenen Bereich des $\mathbb{R}_{\boldsymbol{c}}$ integriert. Mit einer entsprechenden Argumentation, wie sie bei der Minimierung des Risikos (4.1.12) geführt wurde, erhält man

$$\delta^*(\Omega_\kappa\,|\,\boldsymbol{c}) = 1 \qquad \text{wenn} \quad p_\kappa p(\boldsymbol{c}|\Omega_\kappa) = \max_\lambda p_\lambda p(\boldsymbol{c}|\Omega_\lambda)$$

$$\text{und} \quad p_\kappa p(\boldsymbol{c}|\Omega_\kappa) \geq \alpha(\boldsymbol{c}) \ .$$

Dieses entspricht dem ersten Teil von (4.1.25). Damit ist gezeigt, dass bei einer Entscheidung gemäß (4.1.25) die Wahrscheinlichkeit p_{c} einer korrekten Entscheidung maximiert wird. Dieses Ergebnis wird zusammengefasst in

Satz 4.2 *Für die spezielle Kostenfunktion (4.1.19) ergibt die Minimierung des Risikos die Entscheidungsregel (4.1.25). Der damit arbeitende Klassifikator ist identisch mit dem Klassifikator, der bei fester Rückweisungswahrscheinlichkeit die Fehlerwahrscheinlichkeit minimiert.*

Zur Festlegung der Kosten (4.1.19) braucht man also statt der $k(k+1)$ Zahlen $r_{\lambda\kappa}$ in (4.1.4) nur die eine Zahl β in (4.1.27) bzw. p_{z} in (4.1.26) zu wählen. Im Unterschied zu den „Kosten“ $r_{\lambda\kappa}$ sind Rückweisungs- und Fehlerwahrscheinlichkeit unmittelbar anschauliche Größen.

Die $(0,1)$–Kostenfunktion

In manchen Fällen ist eine Rückweisungsmöglichkeit unerwünscht. Wenn stets eine Entscheidung für genau eine der k Klassen Ω_κ getroffen werden soll, ist lediglich Ω_0 auszuschließen. In (4.1.6) ist dieses bereits mit aufgeführt, in (4.1.12) – (4.1.16) darf λ nur die Werte von 1 bis k durchlaufen; dann gilt die optimale Entscheidungsregel (4.1.16) auch bei Ausschluss der Rückweisung, also bei **erzwungener Entscheidung**. Eine in diesem Fall häufig angewendete Wahl der Kosten ist die sogenannte (0,1)–Kostenfunktion

$$\begin{aligned} r_{\kappa\kappa} &= 0 \ , \\ r_{\lambda\kappa} &= 1 \ , \quad \lambda \neq \kappa \ , \quad \kappa, \lambda = 1, \ldots, k \ . \end{aligned} \tag{4.1.30}$$

Das Risiko in (4.1.12) und (4.1.20) reduziert sich auf

$$\boxed{V(\delta) = p_{\mathrm{f}} \ .} \tag{4.1.31}$$

Die Minimierung des Risikos entspricht hier also der Minimierung der Fehlerwahrscheinlichkeit. Für die Prüfgrößen $u_\lambda(\boldsymbol{c})$ in (4.1.13) erhält man

$$u_\lambda(\boldsymbol{c}) = \sum_{\substack{\kappa=1 \\ \kappa\neq\lambda}}^{k} p_\kappa p(\boldsymbol{c}|\Omega_\kappa) \ , \quad \lambda = 1, \ldots, k \ . \tag{4.1.32}$$

Die Entscheidungsregel (4.1.16) ist bei dieser speziellen Kostenfunktion äquivalent der Regel

$$\boxed{\delta^*(\Omega_\kappa|\boldsymbol{c}) = \begin{cases} 1 & : \ p_\kappa p(\boldsymbol{c}|\Omega_\kappa) = \max\limits_{\lambda} p_\lambda p(\boldsymbol{c}|\Omega_\lambda) \ , \\ 0 & : \ \lambda \neq \kappa \ , \quad \lambda = 1, \ldots, k \ . \end{cases}} \tag{4.1.33}$$

Ein Klassifikator, der die Entscheidungsregel (4.1.33) anwendet, wird auch als BAYES-Klassifikator bezeichnet. Die von ihm erreichte Fehlerwahrscheinlichkeit wurde in Abschnitt 3.9 mit p_{B} bezeichnet und ist unten in (4.1.38) angegeben. Es gibt *keinen* Klassifikator, der unter den oben genannten Voraussetzungen eine geringere Fehlerwahrscheinlichkeit als p_{B}

erreicht. Mit (4.1.2) gilt für den BAYES-**Klassifikator** (s. (4.1.3))

$$\boxed{p(\Omega_\lambda|\boldsymbol{c}) = \frac{p_\lambda p(\boldsymbol{c}|\Omega_\lambda)}{p(\boldsymbol{c})}\ .} \tag{4.1.34}$$

Da $p(\boldsymbol{c})$ unabhängig vom Index λ ist, nehmen die **a posteriori Wahrscheinlichkeiten** $p(\Omega_\lambda|\boldsymbol{c})$ für den gleichen Index ihr Maximum an wie die Ausdrücke $p_\lambda p(\boldsymbol{c}|\Omega_\lambda)$ in (4.1.33). Sind zudem die a priori Wahrscheinlichkeiten gleich, so ist die Maximierung von $p(\Omega_\lambda|\boldsymbol{c})$ äquivalent der Maximierung von $p(\boldsymbol{c}|\Omega_\lambda)$. Ein solcher Klassifikator wird als **Maximum-likelihood-Klassifikator** bezeichnet. Eine zur Entscheidungsregel (4.1.33) äquivalente Formulierung besteht darin, den Index κ der ausgewählten Klasse zu berechnen aus

$$\boxed{\kappa = \underset{\lambda\in\{1,\ldots,k\}}{\operatorname{argmax}}\, p(\Omega_\lambda|\boldsymbol{c}) = \underset{\lambda\in\{1,\ldots,k\}}{\operatorname{argmax}}\, p_\lambda p(\boldsymbol{c}|\Omega_\lambda)\ .} \tag{4.1.35}$$

Dieses Ergebnis wird zusammengefasst in

Satz 4.3 *Der* BAYES-*Klassifikator, der bei erzwungener Entscheidung die Fehlerwahrscheinlichkeit minimiert, berechnet die k a posteriori Wahrscheinlichkeiten (4.1.34) und entscheidet sich für die Klasse mit maximaler a posteriori Wahrscheinlichkeit; dieses entspricht der Anwendung der Entscheidungsregel (4.1.33) bzw. (4.1.35).*

Es wird noch darauf hingewiesen, dass sich auch die allgemeinen Prüfgrößen (4.1.13) oder (4.1.18) mit (4.1.34) umformen lassen, sodass bei der Bestimmung des Minimums (4.1.15) nur die a posteriori Wahrscheinlichkeiten verwendet werden. Man kann also wahlweise die Terme $p_\lambda p(\boldsymbol{c}|\Omega_\lambda)$ oder $p(\Omega_\lambda|\boldsymbol{c})$ nehmen, da der normierende Faktor $p(\boldsymbol{c})$ von $p_\lambda p(\boldsymbol{c}|\Omega_\lambda)$ unabhängig ist. Für die numerische Berechnung wird man in der Regel von $p_\lambda p(\boldsymbol{c}|\Omega_\lambda)$ ausgehen, da die a priori Wahrscheinlichkeiten p_λ und die bedingten Dichten $p(\boldsymbol{c}|\Omega_\lambda)$ als bekannt vorausgesetzt wurden bzw. unter geeigneten Annahmen mit einer klassifizierten Stichprobe geschätzt werden können.

4.1.5 Fehlerwahrscheinlichkeit und Kosten

Es wurden zwei Größen zur Beurteilung der Leistungsfähigkeit eines Klassifikators eingeführt, nämlich die Kosten für bestimmte Entscheidungen und die Wahrscheinlichkeit für das Treffen bestimmter Entscheidungen. Die Wahrscheinlichkeiten haben den Vorteil, dass sie unmittelbar anschaulich sind und auch durch Abzählen der verschiedenen Fälle leicht geschätzt werden können. Beispielsweise kann man bei einem automatischen EKG Auswertesystem abzählen, wie oft ein tatsächlich normales EKG irrtümlich als anormal und ein tatsächlich anormales EKG irrtümlich als normal eingestuft wurde – vorausgesetzt, es gibt eine übergeordnete Instanz, welche die richtige Diagnose kennt. Minimierung der Fehlerwahrscheinlichkeit bedeutet, dass man die Summe der mit den a priori Wahrscheinlichkeiten bewichteten beiden Fehlerarten minimiert. In diesem Beispiel wird man aber vermutlich den Fall, dass ein anormales EKG für normal gehalten wird, stärker bewichten wollen, damit er weniger häufig auftritt. Dieses ist durch Zuordnen von Kosten zu den einzelnen Entscheidungen möglich. Andererseits ist es schwierig, solche Kosten konkret in einer Währungseinheit anzugeben, und es müssen unrealistische Kostenzuordnungen vermieden werden. Werden nämlich im obigen Beispiel sehr hohe Kosten der Fehlklassifikation eines anormalen EKG zugeordnet, so wird es für den Klassifikator „am billig-

sten“, fast alle EKG als anormal einzustufen oder zurückzuweisen. Das aber entspricht nicht den Erwartungen an ein nützliches System. Da es meistens schwierig ist, die Kosten als Vielfache irgendeiner Währungseinheit anzugeben, ist es sinnvoller, sie als Gewichtsfaktoren aufzufassen, mit denen man die Häufigkeit bestimmter Entscheidungen erhöhen oder auch erniedrigen kann unter Inkaufnahme einer Erniedrigung oder Erhöhung der Häufigkeit anderer Entscheidungen. Die Häufigkeit der möglichen Entscheidungen ist durch (4.1.8) bestimmbar und hängt über δ und (4.1.16) von den gewählten Kosten ab. Allerdings wird i. Allg. der Zusammenhang zwischen diesen Häufigkeiten und den Kosten nur näherungsweise numerisch berechenbar sein, und dafür ist eine einfache Kostenfunktion wie (4.1.19) besonders geeignet.

Die obige Diskussion zeigt, dass anschaulich wichtige Größen zur Beurteilung eines Klassifikators die Wahrscheinlichkeiten $p(\Omega_\lambda\,|\,\Omega_\kappa)$ sind. Aus (4.1.8) und (4.1.16) folgt

$$p(\Omega_\lambda\,|\,\Omega_\kappa) = \int\limits_{\{\boldsymbol{c}\,|\,u_\lambda(\boldsymbol{c})=\min_j u_j(\boldsymbol{c})\}} p(\boldsymbol{c}\,|\,\Omega_\kappa)\,\mathrm{d}\boldsymbol{c} \quad \text{für} \quad \lambda = 0, 1, \ldots, k \text{ und } \kappa = 1, \ldots, k\ . \tag{4.1.36}$$

Dieses enthält alle i. Allg. möglichen Entscheidungen, um ein Muster aus Ω_κ zu klassifizieren. In dem speziellen Fall der (0,1)-Kostenfunktion ergibt sich aus (4.1.8) und (4.1.16) für die Wahrscheinlichkeit der richtigen Klassifikation von Mustern aus Ω_κ

$$p(\Omega_\kappa\,|\,\Omega_\kappa) = \int\limits_{\{\boldsymbol{c}\,|\,p_\kappa p(\boldsymbol{c}\,|\,\Omega_\kappa)=\max_j p_j p(\boldsymbol{c}\,|\,\Omega_j)\}} p(\boldsymbol{c}\,|\,\Omega_\kappa)\,\mathrm{d}\boldsymbol{c}\ . \tag{4.1.37}$$

Die Wahrscheinlichkeit, mit dem optimalen Klassifikator (4.1.33) Muster aus Ω *falsch* zu klassifizieren, ist also

$$\begin{aligned} p_\mathrm{B} = p_{\mathrm{f,min}} &= 1 - p_\mathrm{c} \\ &= 1 - \sum_{\kappa=1}^{k} p_\kappa \int\limits_{\{\boldsymbol{c}\,|\,p_\kappa p(\boldsymbol{c}\,|\,\Omega_\kappa)=\max_j p_j p(\boldsymbol{c}\,|\,\Omega_j)\}} p(\boldsymbol{c}\,|\,\Omega_\kappa)\,\mathrm{d}\boldsymbol{c} \\ &= 1 - \int\limits_{\mathbb{R}_{\boldsymbol{c}}} \max_{\kappa\in\{1,\ldots,k\}} p_\kappa p(\boldsymbol{c}\,|\,\Omega_\kappa)\,\mathrm{d}\boldsymbol{c}\ . \end{aligned} \tag{4.1.38}$$

Diese Beziehung veranschaulicht Bild 4.1.3. Wenn der optimale Klassifikator realisiert wurde, lässt sich p_B gemäß (3.9.9) schätzen. Der Schätzwert $\widehat{p}_B$ wird auch als *Fehlerrate* bezeichnet. Im Allgemeinen lassen sich nach dieser Methode auch die Wahrscheinlichkeiten $p(\Omega_\lambda\,|\,\Omega_\kappa)$ in (4.1.36) schätzen. Es gilt

$$\widehat{p}(\Omega_\lambda\,|\,\Omega_\kappa) = \frac{\text{Zahl der Muster aus } \Omega_\kappa,\ \text{die } \Omega_\lambda \text{ zugeordnet wurden}}{\text{Zahl der Muster aus } \Omega_\kappa}\ , \tag{4.1.39}$$

wobei natürlich der Klassifikator gemäß der verwendeten Entscheidungsregel zu realisieren ist. Aus (4.1.39) ergeben sich Schätzwerte für die Wahrscheinlichkeit der korrekten Klassifikation $\widehat{p}_\mathrm{c}$, der Rückweisung $\widehat{p}_\mathrm{z}$ und der Fehlklassifikation $\widehat{p}_\mathrm{f}$ für Muster aus dem Problemkreis Ω zu

$$\widehat{p}_\mathrm{c} = \sum_{\kappa=1}^{k} p_\kappa \widehat{p}(\Omega_\kappa\,|\,\Omega_\kappa)\ ,$$

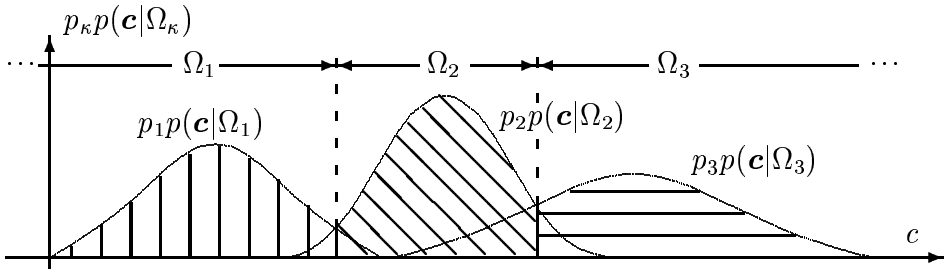


Bild 4.1.3: Muster in dem mit Ω_1 bezeichneten Bereich sind mit einer Wahrscheinlichkeit, die der senkrecht schraffierten Fläche entspricht, tatsächlich aus Ω_1; entsprechendes gilt für Ω_2 usw. Die Wahrscheinlichkeit p_c ein Muster richtig zu klassifizieren, entspricht also der Summe der schraffierten Flächen, und das ist gerade das in (4.1.38) auftretende Integral

$$\begin{aligned} \widehat{p}_\mathrm{z} &= \sum_{\kappa=1}^{k} p_\kappa \widehat{p}(\Omega_0\,|\,\Omega_\kappa)\ , \\ \widehat{p}_\mathrm{f} &= \sum_{\kappa=1}^{k} p_\kappa \sum_{\substack{\lambda=1 \\ \lambda\neq\kappa}}^{k} \widehat{p}(\Omega_\lambda\,|\,\Omega_\kappa)\ , \\ 1 &= \widehat{p}_\mathrm{c} + \widehat{p}_\mathrm{z} + \widehat{p}_\mathrm{f}\ . \end{aligned} \tag{4.1.40}$$

4.1.6 Verallgemeinerungen der Klassifikation eines Merkmalsvektors

Wegen der Problematik der geeigneten Definition allgemeiner Kostenfunktionen $r_{\lambda\kappa}$ wird oft die $(0,1)$–Kostenfunktion verwendet, d. h. die Fehlerwahrscheinlichkeit minimiert. Die Vorgehensweise ist durch den BAYES-Klassifikator und Satz 4.3 gegeben. Dort ist zunächst nur die Klassifikation eines einfachen Musters, das durch einen Merkmalsvektor fester Dimension beschrieben wird, betrachtet worden. Es gibt jedoch einige praktisch wichtige unmittelbare Verallgemeinerungen, die am Beispiel der Klassifikationsprobleme von Abschnitt 1.5 betrachtet werden.

1. Der einfachste und auch der „klassische“ Fall ist die Klassifikation eines *einzelnen Merkmalsvektors* bzw. eines Musters als Ganzes in Bild 1.5.1, S. 30. Es werden gemäß (4.1.34) die a posteriori Wahrscheinlichkeiten aller Klassen berechnet und die Klasse mit maximaler a posteriori Wahrscheinlichkeit ausgewählt

$$\boxed{p(\Omega_\lambda\,|\,\boldsymbol{c}) = \frac{p(\Omega_\lambda)p(\boldsymbol{c}|\Omega_\lambda)}{p(\boldsymbol{c})}\ .} \tag{4.1.41}$$

2. Daraus ergibt sich sofort die allgemeine Lösung für die Klassifikation von *Mustern im Kontext* in Bild 1.5.2, S. 31. Dazu wird die Folge der beobachteten Merkmalsvektoren $[{}^1\boldsymbol{c},\ldots,{}^N\boldsymbol{c}]$ zu einem erweiterten Merkmalsvektor $\boldsymbol{C}$ zusammengefasst und analog die Folge der zu bestimmenden Klassen $[{}^1\Omega,\ldots,{}^N\Omega]$ zu einer erweiterten Klasse $\boldsymbol{\Omega}$. Damit

geht (4.1.34) über in

$$\boxed{p(\boldsymbol{\Omega}|\boldsymbol{C}) = \frac{p(\boldsymbol{\Omega})p(\boldsymbol{C}|\boldsymbol{\Omega})}{p(\boldsymbol{C})}\ .} \tag{4.1.42}$$

Ein wesentliches Problem liegt hier in der *Komplexität* der Bestimmung der maximalen a posteriori Wahrscheinlichkeit. Auf Lösungsansätze dafür wird in Abschnitt 4.7 eingegangen.

3. Die Klassifikation von *Texturen* (Bild 1.5.4, S. 32) bedeutet entweder die Klassifikation eines Musters als Ganzes oder von Mustern im Kontext, bringt also gegenüber den obigen beiden Gleichungen nichts Neues.
4. Die Klassifikation *isoliert gesprochener Wörter* (Worterkennung) basiert direkt auf (4.1.34) . Die Erkennung der Wörter in *zusammenhängend gesprochener Sprache* (kontinuierliche Spracherkennung), die in Bild 1.5.5, S. 32, illustriert wurde, beruht auf einer Verallgemeinerung von (4.1.34). Der üblichen Notation folgend bezeichnen wir das beobachtete akustische Signal mit $\boldsymbol{o}$, wobei es im Augenblick unerheblich ist, ob dieses direkt das Ausgangssignal eines Mikrofons ist, eine Folge von Merkmalsvektoren oder dergleichen. Eine Folge der Länge N von Wörtern aus einem vorgegebenen Vokabular wird mit $\boldsymbol{w} = [w_1, \ldots, w_N]$ bezeichnet. Der BAYES-Ansatz beruht dann wieder darauf, die Wortfolge mit maximaler a posteriori Wahrscheinlichkeit zu bestimmen, d. h.

$$\boxed{p(\boldsymbol{w}\,|\,\boldsymbol{o}) = \frac{p(\boldsymbol{w})p(\boldsymbol{o}\,|\,\boldsymbol{w})}{p(\boldsymbol{o})}\ .} \tag{4.1.43}$$

Diese Beziehung enthält in dem Term $p(\boldsymbol{o}\,|\,\boldsymbol{w})$ die *akustische Information* und in dem Term $p(\boldsymbol{w})$ ein *stochastisches Sprachmodell*, sodass sich eine theoretisch fundierte Lösung auf einer einheitlichen Basis ergibt.

5. Das Problem der *Erkennung dreidimensionaler Objekte* (Objekterkennung) aus zweidimensionalen Ansichten wurde in Bild 1.5.8, S. 33, dargestellt. Hier spielen noch die Algorithmen, die auf der Zuordnung von Modell– zu Bildmerkmalen basieren, eine große Rolle; allerdings gewinnen statistische Ansätze zunehmend Interesse und Bedeutung. Wenn man das statistische Modell der Klasse Ω_κ mit $\mathcal{M}_\kappa$ bezeichnet und die im Bild beobachteten Merkmale mit $\boldsymbol{O}$, so ergibt der BAYES-Ansatz direkt

$$\boxed{p\left(\mathcal{M}_\lambda\,|\,\boldsymbol{O}\right) = \frac{p(\mathcal{M}_\lambda)p\left(\boldsymbol{O}\,|\,\mathcal{M}_\lambda\right)}{p(\boldsymbol{O})}\ .} \tag{4.1.44}$$

In dieser allgemein gehaltenen Formulierung ist z. B. das Problem der unbekannten Lageparameter in $\mathcal{M}_\lambda$ „versteckt“.

Für die Bestimmung der Klasse mit maximaler a posteriori Wahrscheinlichkeit ist in (4.1.44), wie auch in den anderen drei obigen Gleichungen, der Nenner unerheblich.

Man sieht, dass die statistische Entscheidungsthoerie mit ihrem Ansatz der *Minimierung des Risikos* bei der Klassifikation theoretisch fundierte Lösungen für recht unterschiedliche Probleme liefert. Insbesondere lässt sich das Problem der Unterscheidung von *mehreren Klassen*, das

Problem der *Berücksichtigung von Kontext*, das Problem der *Kombination von Informationsquellen* und i. Allg. das Problem des Treffens einer optimalen *Folge von Entscheidungen* durch den BAYES-Klassifikator lösen. Was die statistische Entscheidungsthoerie *nicht leistet* ist die Entwicklung von *statistischen Modellen*, die die Gegebenheiten eines Problemkreises angemessen approximieren, und die Entwicklung von *effizienten Algorithmen* für die Auswertung z. B. von (4.1.43) oder (4.1.44).

Jeder gute Algorithmus zur Klassifikation von Mustern wird versuchen, ein geeignetes Gütekriterium zu optimieren. Die mittleren Kosten bzw. das Risiko V in (4.1.10) sind ein Beispiel für ein relativ allgemeines Gütekriterium, die Fehlerwahrscheinlichkeit p_{f} ist ein Beispiel für ein spezialisiertes Gütekriterium. Für beide wurden in diesem Abschnitt Lösungen angegeben. Weitere Elemente eines Gütekriteriums können z. B. die Komplexität (Rechenzeit), der Trainingsaufwand, die Kosten einer speziellen Hardwarerealisierung, die Adaptierbarkeit an neue Beobachtungen oder die Wartbarkeit sein. Diese Elemente entziehen sich bisher einer formalen Behandlung. Wie schon erwähnt, ist ein anderer Ansatz die Einbeziehung des Einflusses einer endlichen Stichprobe in die Ableitung eines Klassifikators; dafür wird auf Abschnitt 4.3 verwiesen.

Schließlich ist es naheliegend, die *gleiche* Klassifikationsaufgabe mit *mehreren* Klassifikatoren durchzuführen und deren Ergebnisse so zu kombinieren, dass die resultierende Fehlerrate *kleiner* ist als die jedes einzelnen Klassifikators. Im Prinzip liegt hier ein erneutes Klassifikationsproblem vor, das in optimaler Weise zu lösen ist; wie das geschieht, folgt grundsätzlich aus Satz 4.1 – Satz 4.3. Soll also z. B. die aus der Kombination oder **Fusion von Klassifikatoren** resultierende Fehlerrate minimiert werden, so werden die (nichtquantisierten analogen) Ausgaben der Klassifikatoren als Merkmalsvektor eines weiteren BAYES-Klassifikators aufgefasst. Auf diese Weise können prinzipiell ganz verschiedene Klassfikationsergebnisse wie die eines statistischen Klassifikators, einer Support Vektor Maschine und eines neuronalen Netzes kombiniert werden. Da der BAYES-Klassifikator auf recht unterschiedliche Arten approximiert werden kann (wie eben z. B. mit einem statistischen Klassifikator, einer Support Vektor Maschine oder einem neuronalen Netz) ergeben sich hier zahlreiche mögliche Varianten. Für systematische Verfahren zur Generierung unterschiedlicher Klassifikatoren sowie zur Fusion von deren Ergebnissen wird auf die Literatur in Abschnitt 4.11 verwiesen.

4.1.7 Klassenspezifische Klassifikation

Die optimale Entscheidungsregel in (4.1.16) sowie auch speziell der BAYES-Klassifikator in Satz 4.3 sind so aufgebaut, dass *ein und derselbe* Merkmalsvektor für *jede* der k Klassen Ω_κ verwendet wird. Es ist bemerkenswert, dass die statistische Entscheidungstheorie sich auch so formulieren lässt, dass für jede Klasse *klassenspezifische Merkmale* gemäß (3.1.3), S. 164, verwendet werden, was in einer *klassenspezifischen Klassifikation* resultiert. Die Entscheidungsregel (4.1.35) gilt im Prinzip für *beliebige* Merkmalsvektoren, insbesondere natürlich auch für die ursprünglichen Abtastwerte des Musters. Wenn im Merkmalsvektor die gleiche Information enthalten ist wie in den Abtastwerten, gilt

$$\begin{aligned} \kappa &= \operatorname*{argmax}_{\lambda\in\{1,\ldots,k\}} p_\lambda p(\boldsymbol{c}\,|\,\Omega_\lambda) \\ &= \operatorname*{argmax}_{\lambda\in\{1,\ldots,k\}} p_\lambda p(\boldsymbol{f}\,|\,\Omega_\lambda)\ , \end{aligned} \tag{4.1.45}$$

wobei $p(\boldsymbol{c}|\Omega_\lambda)$ und $p(\boldsymbol{f}\,|\Omega_\lambda)$ natürlich verschiedene Funktionen bezeichnen. Der Vorteil der oberen Gleichung liegt darin, dass man i. Allg. versucht, Merkmalsvektoren zu finden, die wesentlich kleiner sind als das gegebene Muster und die fast die gleiche Information enthalten wie dieses.

Bei der klassenspezifischen Klassifikation wird zunächst die bedingte Verteilungsdichte der klassenspezifischen Merkmale bestimmt, d. h. ein Problem in einem niedrigdimensionalen Raum gelöst. Von dieser Verteilungsdichte wird auf die der Abtastwerte zurückgeschlossen, indem sie mit Hilfe des Projektionstheorems in den Raum der Abtastwerte projiziert wird. Im Allgemeinen können *viele* Verteilungsdichten der (hochdimensionalen) Abtastwerte zur gleichen Verteilung der (niedrigdimensionalen) Merkmale führen. Es ist also nur möglich, eine Approximation der tatsächlichen Verteilungsdichte der Abtastwerte zu finden. Das Projektionstheorem erfordert dafür die Verteilungsdichte einer Referenz– oder Nullhypothese. Man kann den ganzen Prozess so interpretieren, dass nicht mehr wie bisher *eine* Menge von Merkmalen bestimmt wird, welche die Klassen möglichst gut *trennen*, sondern *mehrere* Mengen, von denen jede die Muster einer Klasse möglichst gut *beschreibt*.

Als erstes sind also k klassenspezifische Merkmalsvektoren zu wählen, deren bedingte Verteilungsdichten hier wie in (4.1.1) bis auf unbekannte Parameter als bekannt vorausgesetzt werden

$$ {}^{\varrho}\boldsymbol{c}^{(\lambda)} \;=\; T^{(\lambda)}\{\,{}^{\varrho}\boldsymbol{f}\,\} \quad , \qquad p(\boldsymbol{c}^{(\lambda)}\,|\,\Omega_\lambda) = p(\boldsymbol{c}^{(\lambda)}\,|\,\boldsymbol{a}_\lambda)\;. $$

Das Projektionstheorem ist eine Verallgemeinerung der Beziehung zur Variablentransformation von Verteilungsdichten. Wenn eine Zufallsvariable $\boldsymbol{y}$ mit der Verteilungsdichte $p(\boldsymbol{y})$ gegeben ist und wenn $\boldsymbol{y}$ durch eine eineindeutige Transformation $\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{x})$ aus einer Zufallsvariablen $\boldsymbol{x}$ hervorgeht, so kann die Verteilungsdichte von $\boldsymbol{x}$ bestimmt werden aus $p(\boldsymbol{x}) = |\boldsymbol{J}\,|p(\boldsymbol{y}) = |\partial\boldsymbol{\phi}/\partial\boldsymbol{x}\,|p(\boldsymbol{y})$. Ein Merkmalsvektor $\boldsymbol{c}$ hat aber i. Allg. weniger Komponenten als das Muster $\boldsymbol{f}$ Abtastwerte hat, d. h. die Transformation ist nicht eindeutig umkehrbar und man kann nicht eindeutig von der Verteilung $p(\boldsymbol{c})$ der Merkmalsvektoren auf die Verteilungsdichte $p(\boldsymbol{f})$ der Abtastwerte zurückschließen. Das Projektionstheorem ist gerade für diesen Fall der nicht eindeutigen Umkehrbarkeit anwendbar. Voraussetzung für dieses Theorem ist, dass für eine Referenzhypothese H_0 das Paar von Verteilungsdichten $p(\boldsymbol{f}\,|\,H_0)$, $p(\boldsymbol{c}\,|\,H_0)$ bekannt ist und das Verhältnis $p(\boldsymbol{f}\,|\,H_0)/p(\boldsymbol{c}\,|\,H_0)$ für alle Werte von Transformationspaaren $(\boldsymbol{f},\ \boldsymbol{c})$ existiert.

Satz 4.4 *(***Projektionstheorem***) Unter den genannten Voraussetzungen ist*

$$ p(\boldsymbol{f}) = \frac{p(\boldsymbol{f}\,|\,H_0)}{p(\boldsymbol{c}\,|\,H_0)}\,p(\boldsymbol{c}) \tag{4.1.46} $$

eine Verteilungsdichte, deren Integral 1 ist. Werden Werte $\boldsymbol{f}$ mit dieser Verteilungsdichte generiert, so haben daraus gewonnene Merkmale $\boldsymbol{c}$ die Verteilungsdichte $p(\boldsymbol{c})$.
Beweis*: s. z. B. [Baggenstoss, 2001].*

Der Quotient $p(\boldsymbol{f}\,|\,H_0)/p(\boldsymbol{c}\,|\,H_0)$ übernimmt formal die Rolle der JACOBI-Matrix $\boldsymbol{J}$. Wie schon erwähnt ist aus $p(\boldsymbol{c})$ nicht eindeutig auf $p(\boldsymbol{f})$ zu schließen, d. h. (4.1.46) gibt *eine* Verteilungsdichte mit den genannten Eigenschaften, aber nicht die einzige; dieses wird in der Notation jedoch nicht zum Ausdruck gebracht. Von den i. Allg. unendlich vielen in Frage kommenden Verteilungsdichten $p(\boldsymbol{f})$, die unter der (nicht eindeutigen) Merkmalstransformation zur beobachteten Verteilungsdichte $p(\boldsymbol{c})$ der Merkmale führen können, wird mit (4.1.46) diejenige ausgewählt, die das Likelihood–Verhältnis in beiden Repräsentationen konstant hält, d. h. für die

gilt

$$\frac{p(\boldsymbol{f})}{p(\boldsymbol{f}\,|\,H_0)} = \frac{p(\boldsymbol{c})}{p(\boldsymbol{c}|\,H_0)}\ . \tag{4.1.47}$$

Dieses ist ein intuitiv sinnvolles Auswahlkriterium.

Zur Nutzung des obigen Satzes werden k Referenzhypothesen $H_{0,\lambda},\ \lambda = 1,\ldots,k$ gewählt; diese können für jede Klasse verschieden sein, müssen es aber nicht. Bei Wahl nur einer Referenzhypothese für alle k Klassen kann man die klassenspezifische Merkmale so verstehen, dass sie jede Klasse von der gemeinsamen Referenzhypothese *unterscheiden*. Bei Wahl von k verschiedenen Referenzhypothesen kann man die klassenspezifischen Merkmale so verstehen, dass sie jede Klasse *beschreiben* – aus der Beschreibung resultiert dann die Unterscheidung. Weiter werden k klassenspezifische Merkmalsvektoren $\boldsymbol{c}^{(\lambda)}$ gewählt. Aus den Verteilungsdichten der klassenspezifischen Merkmale werden die der Abtastwerte bestimmt zu

$$p(\boldsymbol{f}\,|\,\Omega_\lambda) = \frac{p(\boldsymbol{f}\,|\,H_{0,\lambda})}{p(\boldsymbol{c}^{(\lambda)}\,|\,H_{0,\lambda})} p(\boldsymbol{c}^{(\lambda)}\,|\,\Omega_\lambda)\ . \tag{4.1.48}$$

Der **klassenspezifische Klassifikator** arbeitet dann in Analogie zu (4.1.45) mit der Regel

$$\boxed{\kappa = \underset{\lambda\in\{1,\ldots,k\}}{\operatorname{argmax}} \frac{p(\boldsymbol{f}\,|\,H_{0,\lambda})}{p(\boldsymbol{c}^{(\lambda)}\,|\,H_{0,\lambda})}\, p_\lambda\, p(\boldsymbol{c}^{(\lambda)}\,|\,\Omega_\lambda)\ .} \tag{4.1.49}$$

Mit dieser Formulierung des Klassifikationsproblems werden neue Möglichkeiten und Anforderungen an die stochastische Modellierung eröffnet, die noch umfangreicher theoretischer Ausarbeitung und experimenteller Erprobung bedürfen. Eine Erwartung ist, dass mit klassenspezifischen Klassifikatoren der Umfang der Lernstichprobe zur Schätzung der unbekannten Parameter der Verteilungsdichten kleiner sein kann. Der Grund ist, dass in der Regel die Zahl der Komponenten jedes klassenspezifischen Merkmalsvektors deutlich kleiner sein sollte als die eines für alle Klassen gemeinsamen Merkmalsvektors. Ein weiterer Gesichtspunkt ist die inhärente Modularität der Verarbeitung. Wird nämlich ein Merkmalsvektor in zwei Schritten gewonnen durch $\boldsymbol{c}^{(\lambda)} = T_2^{(\lambda)}\{\boldsymbol{c}^{(\lambda)'}\}$ und $\boldsymbol{c}^{(\lambda)'} = T_1^{(\lambda)}\{\boldsymbol{f}\}$, so kann das Projektionstheorem zweimal angewendet werden und ergibt

$$p(\boldsymbol{f}\,|\,\Omega_\lambda) = \frac{p(\boldsymbol{f}\,|\,H_{0,\lambda})}{p(\boldsymbol{c}^{(\lambda)'}\,|\,H_{0,\lambda})} \frac{p(\boldsymbol{c}^{(\lambda)'}\,|\,H'_{0,\lambda})}{p(\boldsymbol{c}^{(\lambda)}\,|\,H'_{0,\lambda})}\, p(\boldsymbol{c}^{(\lambda)}\,|\,\Omega_\lambda) \tag{4.1.50}$$

Damit lassen sich für mehrere Klassen gemeinsame Schritte zusammenfassen und brauchen nur einmal realisiert zu werden.

4.2 Statistische Klassifikatoren (VA.3.3.4, 29.09.2004)

4.2.1 Statistische Modellierung von Beobachtungen

Die Anwendung statistischer Klassifikationsverfahren setzt die Kenntnis der klassenbedingten Verteilungsdichten $p(\boldsymbol{c}|\Omega_\kappa)$ der Merkmalsvektoren voraus. Verallgemeinert lässt sich sagen, dass man die klassenbedingte Verteilung der beobachteten Daten braucht. Diese wird als **statistisches Modell** bezeichnet. Generell muss ein Modell den Anforderungen genügen:

- Das Modell muss die Realität angemessen repräsentieren. In unserem Fall bedeutet das, dass die statistischen Eigenschaften der Merkmale bzw. Beobachtungen hinreichend genau durch das entwickelte statistische Modell approximiert werden müssen.
- Das Modell muss effizient nutzbar sein. Das bedeutet, dass es Algorithmen zur effizienten Berechnung der a posteriori Wahrscheinlichkeiten in (4.1.41) – (4.1.44), S. 318, geben muss.
- Das Modell sollte weitgehend automatisch aus Beobachtungen bzw. Trainingsdaten konstruiert werden können. Zur Konstruktion gehört i. Allg. die Ermittlung der *Struktur* und die Schätzung der *Parameter* des Modells. Die Strukturermittlung erfolgt entweder „manuell“, d. h. durch Versuch–und–Irrtum, oder durch die in Abschnitt 1.6.10 erwähnten evolutionären Algorithmen. Für die Parameterschätzung liefert die Statistik leistungsfähige und theoretisch fundierte Verfahren.

Anfänglich wurde der statistische Ansatz, von wenigen Ausnahmen abgesehen, auf die Angabe einer Verteilungsdichte für einen Merkmalsvektor beschränkt. Diese Verteilungsdichte war zudem in der Regel eine Normalverteilung. Damit sind die Möglichkeiten der statistischen Modellierung natürlich *nicht* erschöpft. Erweiterungen sind z. B.

- die Verwendung von *Mischungen* von Normalverteilungen, womit auch multimodale Dichten modellierbar sind;
- die Modellierung von *statistischen Abhängigkeiten* durch Angabe einer begrenzten Nachbarschaft in den MARKOV-Zufallsfeldern;
- die Berücksichtigung von im Prinzip *beliebigen* statistischen Abhängigkeiten in den BAYES-Netzwerken;
- die Einführung von *verborgenen*, d. h. von nicht direkt beobachtbaren, Variablen, z. B. zur Erfassung des zeitlichen Zusammenhangs in Wortmodellen oder der durch Projektion verlorengegangenen dritten Dimension bei Bildern;
- die statistische Modellierung von (insbesondere unbekannten) *Zuordnungen* zwischen Merkmalen eines Modells und den Merkmalen eines Bildes von einem Objekt;
- die Einbeziehung von *Koordinatentransformationen* in die Verteilungsdichte und die Einführung *lokaler Modellierung* in den statistischen Objektmodellen;
- die Entwicklung leistungsfähiger Algorithmen zur *Schätzung* der unbekannten Parameter in Modellen mit den genannten Erweiterungen.

In diesem Abschnitt wird eine kurze zusammenfassende Übersicht über Ansätze zur stochastischen Modellierung von Beobachtungen gegeben. Für Einzelheiten wird auf die Abschnitte verwiesen, in denen das entsprechende Klassifikationsproblem erörtert wird. So werden in diesem Abschnitt zwar z. B. MARKOV-Modelle erwähnt, die Einzelheiten jedoch im Abschnitt über Worterkennung gebracht. Da ein Modell einen Problemkreis erfassen soll, ist es nicht verwunderlich, dass für unterschiedliche Problemkreise unterschiedliche statistische Modelle entwickelt wurden und werden.

Verteilungsdichte von Merkmalsvektoren

Der einfachste Fall der Klassifikation besteht darin, dass ein Muster als Ganzes einer von k Klassen zugeordnet werden soll und nur dieses Muster, nicht aber weitere „Hintergrundobjekte“, durch den Sensor aufgenommen wird. Der Standardansatz dafür ist, einen (globalen) Merkmalsvektor aus dem aufgenommenen Muster zu extrahieren. Dieses kommt auch in Definition 1.6, S. 15, zum Ausdruck. Aus Satz 4.1, S. 310, geht hervor, dass der statistisch optimale Klassifikator die Kenntnis der klassenbedingten Dichten der Merkmalsvektoren voraussetzt. Da diese in der Regel bei konkreten Problemen nicht gegeben sind, müssen sie mit Hilfe einer Stichprobe von Mustern geschätzt werden. Es wird hier nur der Fall betrachtet, dass die Stichprobe *klassifiziert* ist, also von jedem Muster ${}^{\varrho}\boldsymbol{f}(\boldsymbol{x}) \in \omega$ ist auch die richtige Klasse bekannt; dieses wird auch als **überwachtes Lernen** (s. Abschnitt 4.8.1) bezeichnet. Da eine Zerlegung

$$\omega = \{\omega_1, \omega_2, ..., \omega_k\}\ , \quad \omega_\kappa \subset \Omega_\kappa \tag{4.2.1}$$

der Stichprobe ω in Teilmengen ω_κ gegeben ist, genügt es, die Ermittlung von $p(\boldsymbol{c}|\,\Omega_\kappa)$ mit Hilfe von ω_κ zu betrachten. Zur Bestimmung der Dichten sind folgende Methoden gebräuchlich, die in (4.2.2) zusammengefasst sind:

1. Vorgabe einer parametrischen Familie $\widetilde{p}(\boldsymbol{c}|\,\boldsymbol{a})$ von Verteilungsdichten, Modellierung der klassenbedingten Verteilungsdichte von Merkmalsvektoren durch eine Funktion aus dieser Familie und Schätzung der unbekannten Parameter aus einer klassifizierten Stichprobe.
2. Vorgabe einer parametrischen Familie wie oben, jedoch Modellierung der klassenbedingten Verteilungsdichte von Merkmalsvektoren durch eine Mischungsverteilung, deren Elemente Funktionen aus dieser Familie sind, und Schätzung der unbekannten Parameter.
3. Zur Vereinfachung der Schätzung können Annahmen über statistische Unabhängigkeiten der Komponenten des Merkmalsvektors getroffen werden, bis hin zur klassenweisen statistischen Unabhängigkeit aller Komponenten.

$$p(\boldsymbol{c}|\,\Omega_\kappa) = \begin{cases} p(\boldsymbol{c}|\,\boldsymbol{a}_\kappa) & : \text{ allgemeiner Fall} \\ \prod_{\nu=1}^{n} p(c_\nu|\,\boldsymbol{a}_\kappa) & : \text{ stat. Unabhängigkeit} \\ \mathcal{N}(\boldsymbol{c}|\,\boldsymbol{\mu}_\kappa, \boldsymbol{\Sigma}_\kappa) & : \text{ Normalverteilung} \\ \sum_{l=1}^{L} p_l\, \mathcal{N}(\boldsymbol{c}|\,\boldsymbol{\mu}_{\kappa,l}, \boldsymbol{\Sigma}_{\kappa,l}) & : \text{ Mischungsverteilung} \\ \mathcal{H}(l_1, \ldots, l_n|\,\boldsymbol{c}) & : \text{ Histogramm} \\ l_\nu = 1, \ldots, L_\nu,\ \nu = 1, \ldots, n & \end{cases} \tag{4.2.2}$$

Das Histogramm (s. Abschnitt 2.2.2 und (2.2.2), S. 78) ist eine spezielle nichtparametrische Schätzung der Verteilungsdichte. Auf nichtparametrische Schätzungen wird im Abschnitt 4.2.6 eingegangen. Der Vorteil einer parametrischen Schätzung liegt vor allem darin, dass *alle Information* einer *beliebig großen* Stichprobe in dem Parametervektor $\boldsymbol{a}$ mit *fester Anzahl* von Komponenten enthalten ist. Eine Vergrößerung der Stichprobe vergrößert also nicht die Zahl der zu speichernden Parameter, sondern liefert nur genauere Schätzwerte.

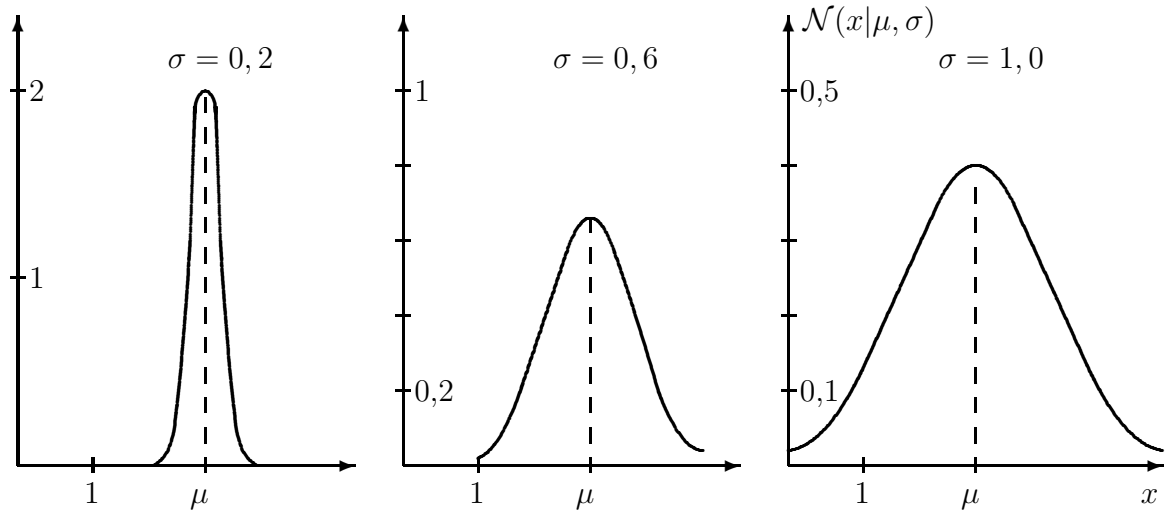


Bild 4.2.1: Beispiele für Normalverteilungen $\mathcal{N}(x\,|\,\mu,\sigma)$

Normalverteilung

Bei der Vorgabe einer parametrischen Familie ist zu beachten, dass hier vor allem solche von Interesse sind, die auch für n–dimensionale Merkmalsvektoren anwendbar sind. Die wichtigste parametrische Familie ist die der n–dimensionalen Normalverteilungen. Beispiele für eindimensionale Normalverteilungen zeigt Bild 4.2.1. Man sieht, dass die Normalverteilung eine **unimodale Verteilungsdichte** ist, d. h. sie hat nur *ein* relatives Maximum. Die eindimensionale Form wurde bereits in (2.1.8), S. 64, eingeführt.

Definition 4.3 *Die n–dimensionale* **Normalverteilung** *bzw. die* GAUSS*-Verteilung ist definiert durch*

$$\boxed{\begin{aligned} p(\boldsymbol{c}|\Omega_\kappa) &= \mathcal{N}(\boldsymbol{c}|\,\boldsymbol{\mu}_\kappa, \boldsymbol{\Sigma}_\kappa) \\ &= \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_\kappa|}} \exp\left[-\frac{1}{2}\,(\boldsymbol{c}-\boldsymbol{\mu}_\kappa)^\mathsf{T}\boldsymbol{\Sigma}_\kappa^{-1}(\boldsymbol{c}-\boldsymbol{\mu}_\kappa)\right] . \end{aligned}} \tag{4.2.3}$$

Jede der Dichten ist vollständig bestimmt durch den Parameter $\boldsymbol{a}_\kappa = (\boldsymbol{\mu}_\kappa, \boldsymbol{\Sigma}_\kappa)$, also den bedingten **Mittelwertvektor**

$$\boldsymbol{\mu}_\kappa = E\{\boldsymbol{c}|\Omega_\kappa\} = \int_{\mathbb{R}_{\boldsymbol{c}}} \boldsymbol{c}\, p(\boldsymbol{c}|\Omega_\kappa)\,\mathrm{d}\boldsymbol{c} \tag{4.2.4}$$

und die bedingte **Kovarianzmatrix**

$$\boldsymbol{\Sigma}_\kappa = E\{(\boldsymbol{c}-\boldsymbol{\mu}_\kappa)(\boldsymbol{c}-\boldsymbol{\mu}_\kappa)^\mathsf{T}\,|\,\Omega_\kappa\} . \tag{4.2.5}$$

Setzt man vereinfachend $\boldsymbol{\Sigma}_\kappa = \mathrm{diag}\left(\sigma_{\kappa,1}^2, \ldots, \sigma_{\kappa,n}^2\right)$, so reduziert sich (4.2.3) zu

$$p(\boldsymbol{c}|\Omega_\kappa) = \frac{1}{\sqrt{\prod_{\nu=1}^{n}(2\pi\sigma_{\kappa,\nu}^2)}} \exp\left[-\frac{1}{2}\sum_{\nu=1}^{n}\left(\frac{c_\nu - \mu_{\kappa,\nu}}{\sigma_{\kappa,\nu}}\right)^2\right] . \tag{4.2.6}$$

Vereinfacht man weiter zu $\boldsymbol{\Sigma}_\kappa = \sigma_\kappa^2 \boldsymbol{I}$, so ergibt sich

$$p(\boldsymbol{c}|\Omega_\kappa) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left[-\frac{1}{2\sigma^2}\sum_{\nu=1}^{n}(c_\nu - \mu_{\kappa,\nu})^2\right] \quad . \tag{4.2.7}$$

Die Zulässigkeit solcher Vereinfachungen ist von Fall zu Fall zu prüfen.

Die Maximum-likelihood-Schätzwerte (s. u.) von Mittelwert und Kovarianzmatrix einer Normalverteilung sind

$$\boldsymbol{\mu}_\kappa \simeq \widehat{\boldsymbol{\mu}}_\kappa = \frac{1}{N_\kappa}\sum_{j=1}^{N_\kappa} {}^j\boldsymbol{c}_\kappa \ , \quad {}^j\boldsymbol{c}_\kappa \in \omega_\kappa \ , \tag{4.2.8}$$

$$\boldsymbol{\Sigma}_\kappa \simeq \widehat{\boldsymbol{\Sigma}}_\kappa = \frac{1}{N_\kappa}\sum_{j=1}^{N_\kappa}({}^j\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_\kappa)({}^j\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_\kappa)^\mathsf{T} \ . \tag{4.2.9}$$

Im Folgenden wird vielfach nicht zwischen den durch (4.2.4) und (4.2.5) definierten Größen und ihren mit (4.2.8) und (4.2.9) berechneten Schätzwerten unterschieden. Die zuverlässige Schätzung der Kovarianzmatrix erfordert einen Stichprobenumfang von etwa $N_\kappa = 1000$ bis 10.000 Mustern je Klasse. Wenn eine klassifizierte Stichprobe ω gemäß (4.2.1) mit N_κ Elementen ${}^j\boldsymbol{c}_\kappa \in \omega_\kappa, j = 1, \ldots, N_\kappa, \kappa = 1, \ldots, k$ gegeben ist, bereitet die Berechnung der Schätzwerte kein Problem. Mit den ermittelten Schätzwerten wird so gerechnet als seien sie die richtigen Werte.

Andere Verteilungen

Es gibt nur wenige andere n–dimensionale parametrische Familien von Dichten, wie die t–Verteilung, die DIRICHLET-Verteilung und die multinomiale Verteilung. Dazu kommen die n–dimensionalen Erweiterungen von eindimensionalen Funktionen sowie die sphärisch symmetrischen Verteilungen. Ist $p(c) = \alpha_1 f(c)$ eine eindimensionale Dichte mit der normierenden Konstante α_1, so ist

$$p(\boldsymbol{c}) = \alpha_n |\boldsymbol{W}|^{1/2} f\left(\left[(\boldsymbol{c}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{W}(\boldsymbol{c}-\boldsymbol{\mu})\right]^{1/2}\right) \tag{4.2.10}$$

eine n–dimensionale Erweiterung. Die Matrix $\boldsymbol{W} = \beta\boldsymbol{\Sigma}^{-1}$ ist durch die Kovarianzmatrix gegeben. Die Konstanten α_n und β sind so zu wählen, dass das Integral über $p(\boldsymbol{c})$ den Wert Eins hat. Da alle Dichten vom Typ (4.2.10) unimodal sind und eine quadratische Form wie in (4.2.3) enthalten, ergeben sie nur eine geringe Verallgemeinerung der Normalverteilungen.

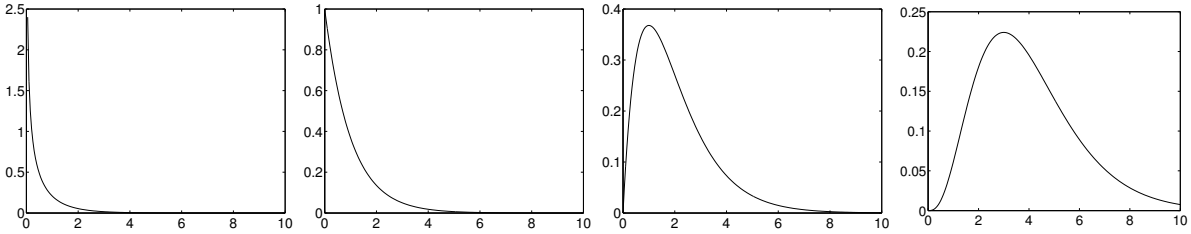
Wir erwähnen weiterhin die n-dimensionalen Exponentialverteilungen

$$p(\boldsymbol{c}|\boldsymbol{a}) = \begin{cases} (\prod_{\nu=1}^n a_\nu)\exp\left[-\boldsymbol{a}^\mathsf{T}\boldsymbol{c}\right] & : \ c_\nu > 0\,, \ \nu = 1, \ldots, n \\ 0 & : \ \text{sonst} \end{cases} \ , \tag{4.2.11}$$

wobei $\boldsymbol{a} = (a_1, \ldots, a_n)^\mathsf{T}\,, \ a_\nu > 0\,, \ \nu = 1, \ldots, n$ ein Parametervektor ist, sowie die eindimensionale Gammaverteilung

$$p(c|\beta,\gamma,\mu) = \frac{1}{\beta\Gamma(\gamma)}\left(\frac{x-\mu}{\beta}\right)^{\gamma-1}\exp\left[-\frac{x-\mu}{\beta}\right] \ , \quad x \geq \mu;\ \beta,\gamma > 0 \tag{4.2.12}$$

mit dem Lageparameter μ, dem Formparameter γ und dem Skalierungsparameter β; eine n–dimensionale Verallgemeinerung beruht auf der Annahme statistischer Unabhängigkeit der Komponenten des Merkmalsvektors wie in (4.2.13). Die standard Gammaverteilung hat die Parameter $\mu = 0$, $\beta = 1$. Der Einfluss des Formparameters γ geht aus Bild 4.2.2 hervor.



$\gamma = 0,5$ $\gamma = 1,0$ $\gamma = 2,0$ $\gamma = 4,0$

Bild 4.2.2: Standard Gammaverteilung für einige Werte des Formparameters γ

Statistische Unabhängigkeiten

Wenn man *klassenweise statistische Unabhängigkeit* der Merkmale annimmt, gilt

$$p(\boldsymbol{c}|\Omega_\kappa) = \prod_{\nu=1}^{n} p(c_\nu|\Omega_\kappa) \ . \tag{4.2.13}$$

Es genügt also, die n *ein*dimensionalen Dichten der Merkmale c_ν zu schätzen. Neben der oben für n-dimensionale Dichten geschilderten Vorgehensweise kommt dafür auch die Schätzung der Dichte mit einem Histogramm (s. auch Abschnitt 2.2.2) in Frage. Die Speicherung der Dichte erfordert dann die Speicherung der n Histogramme. Mit einem Histogramm können auch multimodale Dichten geschätzt und statistische Abhängigkeiten erster Ordnung berücksichtigt werden.

Im Allgemeinen gibt es viele Übergänge von vollständiger statistischer Abhängigkeit bis zu vollständiger statistischer Unabhängigkeit, da jeweils einige Komponenten des Merkmalsvektors von anderen abhängig bzw. unabhängig sein können. Einige Beispiele sind in (4.2.14) für einen vierdimensionalen Merkmalsvektor $\boldsymbol{c} = (c_1, c_2, c_3, c_4)^\mathsf{T}$ gezeigt; dort wurde die Abhängigkeit von der Klasse Ω_κ fortgelassen.

$$p(\boldsymbol{c}) = \begin{cases} p(c_1, c_2, c_3, c_4) & : \text{ vollst. Abhängigkeit} \\ p(c_1)p(c_2|c_1)p(c_3|c_2, c_1)p(c_4|c_3, c_2, c_1) & : \text{ vollst. Abhängigkeit} \\ p(c_1)p(c_2|c_1)p(c_3|c_2, c_1)p(c_4|c_3, c_2) & : \text{ Abh. 2. Ordnung} \\ p(c_1)p(c_2|c_1)p(c_3|c_2)p(c_4|c_3) & : \text{ Abh. 1. Ordnung} \\ p(c_3)p(c_4|c_3)p(c_1|c_4)p(c_2|c_1) & : \text{ Abh. 1. Ordnung} \\ p(c_1)p(c_2)p(c_3)p(c_4) & : \text{ vollst. Unabhängigkeit} \end{cases} \tag{4.2.14}$$

Das Problem ist offenbar, die tatsächlich in der Stichprobe ω vorhandenen statistischen Abhängigkeiten zu erfassen und in die Modellierung von $p(\boldsymbol{c})$ einzubeziehen. Eine formale und effiziente Erfassung von Unabhängigkeiten wird in den unten erwähnten BAYES-Netzen vorgenommen.

Mischungsverteilung

Die in (4.2.2) genannten Mischungsverteilungen lassen sich im Prinzip mit beliebigen parametrischen Familien von Dichten bilden. Praktisch sinnvoll sind nur solche, die im Sinne von Satz 4.19, S. 415, *identifizierbar* sind; dazu gehören Mischungen von Normalverteilungen, die in der Mustererkennung besonders wichtig sind.

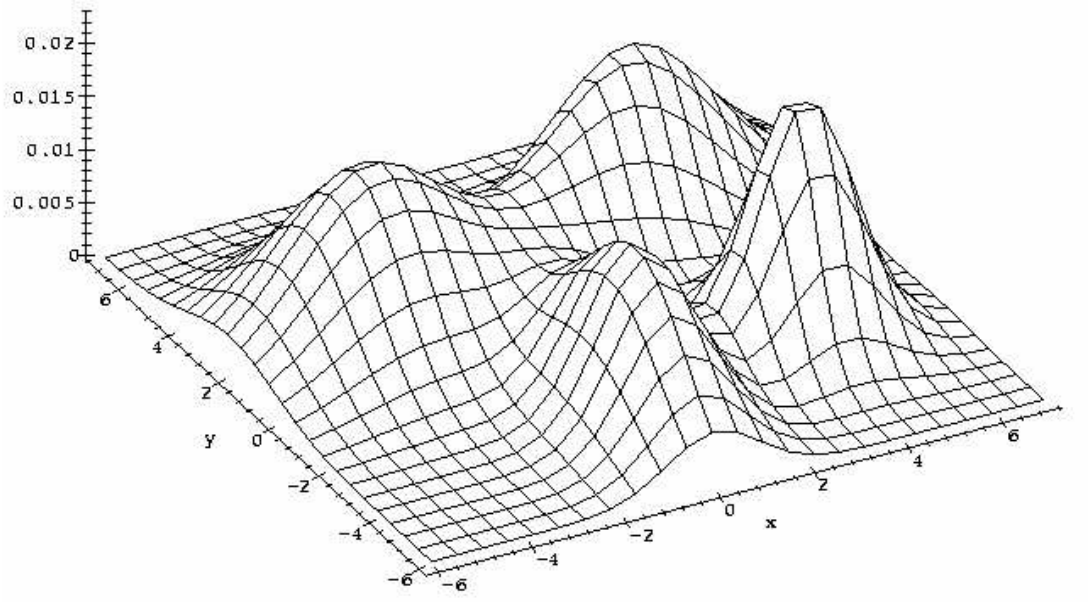


Bild 4.2.3: Mischung von vier zweidimensionalen Normalverteilungen

Definition 4.4 *Eine* **Mischung von Normalverteilungen** *ist gegeben durch*

$$
\begin{aligned}
p(\boldsymbol{c}|\Omega_\kappa) &= \sum_{l=1}^{L_\kappa} p_{\kappa,l}\,\mathcal{N}(\boldsymbol{c}|\boldsymbol{\mu}_{\kappa,l},\boldsymbol{\Sigma}_{\kappa,l})\;, \quad \sum_l p_{\kappa,l} = 1\;, \\
&= \sum_{l=1}^{L_\kappa} p_{\kappa,l}\,\frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_{\kappa,l}|}}\exp\left[-\frac{1}{2}(\boldsymbol{c}-\boldsymbol{\mu}_{\kappa,l})^{\mathsf{T}}\boldsymbol{\Sigma}_{\kappa,l}^{-1}(\boldsymbol{c}-\boldsymbol{\mu}_{\kappa,l})\right]\;. \qquad (4.2.15)
\end{aligned}
$$

Damit sind praktisch beliebige Verteilungsdichten, auch multimodale Dichten, approximierbar. Bei festgelegter Zahl L_κ von Mischungskomponenten je Klasse sind die unbekannten Parameter $\{p_{\kappa,l}, \boldsymbol{\mu}_{\kappa,l}, \boldsymbol{\Sigma}_{\kappa,l}|l = 1,\ldots,L_\kappa\}$. Ein Beispiel einer Mischungsverteilung zeigt Bild 4.2.3. Die Zahl der zu schätzenden Parameter lässt sich oft durch die Vereinfachungen (4.2.6) bzw. (4.2.7) reduzieren, selbst wenn dadurch zur Approximation einer gegebenen Verteilungsdichte eine größere Zahl L_κ von Mischungskomponenten erforderlich wird.

Die obige Mischungsverteilungsdichte wurde zur Modellierung *einer* Klasse eingeführt. Wenn man *keine* klassifizierte Stichprobe vorliegen hat, wird die beobachtete Stichprobe ω des Problemkreises Ω ebenfalls durch eine Mischungsverteilungsdichte beschrieben, die nun aber *alle* k Klassen mischt. Dieser Fall wird in Abschnitt 4.8.2 unter dem Aspekt des *unüberwachten Lernens* betrachtet. Eine zentrale Frage ist, ob bzw. unter welchen Bedingungen die Parameter der Mischungsverteilungsdichte *identifizierbar* sind, d. h. eindeutig aus einer Stichprobe geschätzt werden können. Darauf wird in Abschnitt 4.8.2 eingegangen; die (iterative) Schätzung der Parameter wird in Abschnitt 4.8.3 behandelt.

Histogramm

Ein *Histogramm* ist eine *nichtparametrische Schätzung* der Verteilungsdichte; im Zusammenhang mit dem Grauwerthistogramm wurde darauf bereits in Abschnitt 2.2.2 eingegangen. Wie in (4.2.2) angedeutet, wird jede Komponente c_ν des Merkmalsvektors in L_ν, in der Regel gleich-

große, Intervalle eingeteilt. Für einen n–dimensionalen Merkmalsvektor werden damit $\prod_{\nu=1}^{n} L_\nu$ Hyperquader im $\mathbb{R}^n$ gebildet. Im i–ten Hyperquader wird die Zahl N_i der dort beobachteten Werte des Merkmalsvektors aus einer Stichprobe ω vom Umfang N Muster abgezählt. Der Wert

$$P_i(\boldsymbol{c}) \approx \widehat{P}_i(\boldsymbol{c}) = \frac{N_i}{N} \tag{4.2.16}$$

ist ein Schätzwert der Wahrscheinlichkeit, einen Wert des Merkmalsvektors im i-ten Hyperquader zu beobachten. Division durch das Volumen gibt einen Schätzwert der Dichte. Wenn man zusätzlich klassenweise statistische Unabhängigkeit der Komponenten des Merkmalsvektors annimmt, betrachtet man das i_ν–te Intervall für die Komponente c_ν des Merkmalsvektors und bestimmt die Zahl N_{i_ν} der beobachteten Werte. Der Wert

$$P_i(c_\nu) \approx \widehat{P}_i(c_\nu) = \frac{N_{i_\nu}}{N} \tag{4.2.17}$$

ist ein Schätzwert der Wahrscheinlichkeit, einen Wert der Komponente c_ν im Intervall i_ν zu finden.

Der Vorteil des Histogramms besteht darin, dass keinerlei Annahme über die parametrische Form der Verteilung erforderlich ist. Der Nachteil ist die exponentiell wachsende Zahl der Hyperquader, die diesen Ansatz auf kleine Werte von n beschränkt. Dieses führt schon bei moderaten Zahlen zu vielen leeren Hyperquadern. Nimmt man z. B. $n = 10$ Merkmale, $L_\nu = 10$ Intervalle je Komponente und eine Stichprobe vom Umfang $N = 10^6$ Muster an, so enthalten mindestens $10^{10} - 10^6 = 0,9999 \cdot 10^{10}$ Hyperquader, d. h. fast alle, *keine* Beobachtung; das ist „der Fluch der hohen Dimension“.

Gebräuchlich ist der Ansatz z. B. für Grauwert– bzw. Farbhistogramme, mit $n = 1$ bzw. $n = 3$ (s. Abschnitt 2.2.2), oder gemäß (4.2.17) zusammen mit der Unabhängigkeitsannahme, da dann nur n eindimensionale Histogramme zu schätzen sind.

Modelle mit MARKOV–Zufallsfeldern

Der Wert eines Bildpunktes hängt in der Regel vor allem von den Werten unmittelbar benachbarter Bildpunkte ab und weniger von denen weiter entfernter. Dieses gilt für Grau–, Farb– und Multispektralwerte, d. h. sowohl für Bildpunkte mit skalaren als auch vektoriellen Werten. Wenn man einen multispektralen Bildpunkt klassifiziert, z. B. nach Klassen wie „Wald“, „Wasser“, usw. wird die Klasse sich i. Allg. nicht von Punkt zu Punkt ändern, sondern in einer Nachbarschaft relativ homogen sein. Entsprechendes gilt für die Tiefe oder Geschwindigkeit, die man benachbarten Bildpunkten zuordnet. Da Bildpunkte praktisch ausschließlich in einem rechteckigen Gitter angeordnet sind, wählt man als Nachbarschaft oft die 4– oder 8–Nachbarschaft nach Definition 2.13, S. 136.

Das Standardmodell für die Berechnung von Wahrscheinlichkeiten in solchen Konfigurationen ist das MARKOV-*Zufallsfeld*. Für die Definition im Zusammenhang mit der Klassifikation von Mustern betrachten wir eine Menge von Zufallsvariablen X, deren Werte z. B. Grauwerte f_{jk}, daraus extrahierte Merkmale c_ν, oder Farbe, Bewegung und Tiefe an einem Bildpunkt sein können. Zur Vereinfachung der Notation verwenden wir für die allgemeine Definition nur eine eindimensionale Indizierung und betrachten daher die Menge der Werte

$$\widetilde{X} = \{X_\nu | \nu = 1, \ldots, n\} \tag{4.2.18}$$

von Zufallsvariablen. Für die i–te Zufallsvariable wird, ähnlich wie in (2.4.6), S. 109, eine *Nachbarschaft* $N_m(X_i)$, die $m \leq n$ Werte aus $\widetilde{X}$ enthält, definiert mit

$$N_m(X_i) = \{X_\nu \,|\, X_\nu \text{ ist Nachbar von } X_i\} \; . \tag{4.2.19}$$

Die Nachbarschaft von X_i kann im Prinzip beliebig definiert werden; Beispiele sind die Menge der Bildpunkte, die in einer 4–Nachbarschaft (s. Definition 2.13, S. 136) liegen; oder die Menge der Bildpunkte, deren Grau–, Farb–, Geschwindigkeits– oder Tiefenwerte sich um weniger als ein vorgegebener Schwellwert unterscheiden (die Menge dieser Bildpunkte wird i. Allg. nicht räumlich benachbart oder zusammenhängend sein); oder die Menge der z. B. drei Merkmale $(c_\nu, c_{\nu+1}, c_{\nu+2})$ einer orthonormalen Entwicklung, die aufeinanderfolgende Ordnung haben.

Definition 4.5 *Ein* MARKOV-**Zufallsfeld** *ist definiert durch die Positivität, d. h.* $p(\widetilde{X}) > 0$ *und die* MARKOV-*Eigenschaft*

$$p(X_i \,|\, \{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n\}) = p(X_i \,|\, N_m(X_i)) \; . \tag{4.2.20}$$

Die durch die Menge $\{\widetilde{X} \backslash X_i\}$ bedingte Wahrscheinlichkeit von X_i hängt damit nur von den Elementen in der Nachbarschaft ab. Die MARKOV-Eigenschaft *begrenzt* also die statistischen Abhängigkeiten zwischen Zufallsvariablen umso mehr, je kleiner die Zahl m der Elemente der Nachbarschaft gegenüber der Zahl n der Zufallsvariablen ist. Die MARKOV-Eigenschaft führt dazu, dass sich die Berechnung der Verbundwahrscheinlichkeit $p(\widetilde{X})$ wesentlich vereinfachen lässt.

Die Zufallsvariablen werden nun als *Knoten* eines *Graphen* $\mathcal{G}$ aufgefasst. Zwischen zwei Knoten wird eine *Kante* gezogen, wenn die zugehörigen Zufallsvariablen benachbart sind. Als *Clique* C_i eines Graphen bezeichnet man eine Menge von Knoten, wenn es zwischen jedem Paar von Knoten eine Kante gibt. Die Cliquen sind also vollständig zusammenhängende Teilgraphen. Die Menge aller Cliquen von $\mathcal{G}$ ist $\widetilde{C}_\mathcal{G} = \{C_1, \ldots, C_P\}$ und X_{C_i} ist die Menge der zu der Clique C_i gehörigen Zufallsvariablen. Für die Verbundwahrscheinlichkeit $p(\widetilde{X})$ gelten folgende Aussagen:

Satz 4.5 *Es gibt eine Menge von reellwertigen* **Cliquenfunktionen** $\Phi_{C_i}(X_{C_i}), i = 1, \ldots, P$, *die symmetrisch in den Argumenten sind, sodass gilt*

$$p(\widetilde{X}) = \frac{1}{Z} \prod_{C_i \in \widetilde{C}_\mathcal{G}} \Phi_{C_i}(X_{C_i}) \; . \tag{4.2.21}$$

Dabei ist Z eine Konstante, die die Verteilungsdichte $p(\widetilde{X})$ auf das Integral Eins normiert. Symmetrie in den Argumenten bedeutet, dass eine Vertauschung der Reihenfolge der Argumente keine Änderung des Funktionswertes bewirkt. Die obige Gleichung lässt sich in äquivalenter Form durch ein GIBBS-Feld darstellen:

Satz 4.6 *Die Verbundwahrscheinlichkeit ist auch gegeben durch ein* GIBBS-**Feld**

$$p(\widetilde{X}) \;=\; \frac{1}{Z} \exp\left[- \sum_{C_i \in \widetilde{C}_\mathcal{G}} V_{C_i}(X_{C_i}) \right] = \frac{1}{Z} \exp\Big[-U(\widetilde{X}) \Big] \; . \tag{4.2.22}$$

Die oben eingeführten Funktionen V werden als *Potentialfunktionen* bezeichnet, die Summe über die Potentialfunktionen, nämlich die Funktionen U, werden als *Energiefunktionen* bezeichnet. Die normierende Konstante erhält man durch Summation über die (diskreten) Werte der Zufallsvariablen zu

$$Z = \sum_{\widetilde{X}} \exp\Big[-U(\widetilde{X})\Big] \ . \tag{4.2.23}$$

Damit ist kurz skizziert, dass sich bei geeigneter Beschränkung der statistischen Abhängigkeiten geschlossene und einfache Gleichungen für die Verbundwahrscheinlichkeit einer Menge von Zufallsvariablen ergeben.

Statistische Wortmodelle

Ein wesentliches Kennzeichen gesprochener Sprache ist die zeitliche Aufeinanderfolge von Lauten. Es kommt zwar vor, dass ein oder mehrere Laute *nicht gesprochen* werden, aber die zeitliche Reihenfolge zweier Laute wird vom Sprecher *nicht umgeordnet*, es wird also z. B. nicht statt „morgen“ die Lautfolge „omrgen“ gesprochen. Man braucht daher für die Modellierung von Wörtern ein Modell, das eine zeitliche Reihenfolge definiert und die typischerweise beim Sprechen auftretenden Flüchtigkeiten zulässt. Für die Erkennung sowohl isoliert als auch zusammenhängend gesprochener Wörter haben sich hier (zumindest z. Z.) die Hidden-MARKOV-Modelle durchgesetzt. Im Prinzip handelt es sich um endliche stochastische Automaten. Die Vorstellung ist, dass die zu einem gesprochenen Wort gehörige Folge von Wortuntereinheiten, z. B. von Lauten, durch *zwei* Mechanismen generiert wird, die einen stochastischen Prozess erzeugen.

Der *erste Mechanismus* generiert über der Zeit eine Zustandsfolge $\boldsymbol{s} = [s_n]$ der Länge T. Jeder *Zustand* s_n ist ein Element einer endlichen Menge S von Zustandssymbolen S_i

$$S = \{S_1, S_2, \ldots, S_I\} \ . \tag{4.2.24}$$

Zur diskreten Zeit t_n befindet sich das Modell im Zustand $s_n \in S$. Ein neuer Zustand wird mit einer Wahrscheinlichkeit angenommen, die durch die Matrix der **Zustandsübergangswahrscheinlichkeiten** definiert ist

$$\begin{aligned} \boldsymbol{A}^{I\times I} &= (a_{ij}) \ , \quad i,j = 1,2,\ldots,I \ , \\ a_{ij} &= P\left(s_{n+1} = S_j \,|\, s_n = S_i\right) \ , \quad 0 \le a_{ij} \ , \quad \sum_j a_{ij} = 1 \ . \end{aligned} \tag{4.2.25}$$

Der auf diese Weise generierte **stochastische Prozess** heißt *kausal*, da die Wahrscheinlichkeit für die Generierung eines Zustandes nur von *vergangenen* Zuständen abhängt; er heißt *einfach*, da nur *ein* vorangehender Zustand Einfluss hat; und er heißt *stationär*, weil die Übergangswahrscheinlichkeiten *unabhängig* von der Zeit sind. Auf diese Weise kann eine im Prinzip beliebig lange Folge von Zuständen aus einer beliebigen (nichtleeren) Menge von Zuständen generiert werden. Zur Modellierung eines Wortes werden als Zustände die Wortuntereinheiten (z. B. die Laute) einer Standardaussprache (z. B. nach Duden) verwendet. Ein Zustandsübergang kann z. B. zum nächsten Zustand oder auch zum gleichen wie bisher erfolgen. Damit wird die unterschiedliche *Dauer* von gesprochenen Wörtern modelliert. Man bezeichnet einen solchen stochastischen Prozess $\boldsymbol{s}$ als MARKOV-**Kette**.

Der *erste Zustand* s_1 wird mit einer Wahrscheinlichkeit gewählt, die durch den Vektor der **Anfangswahrscheinlichkeiten** definiert ist

$$\begin{aligned} \boldsymbol{\pi} &= (\pi_i)\ , \quad i = 1, 2, \ldots, I\ , \\ \pi_i &= P(s_1 = S_i)\ , \quad 0 \leq \pi_i\ , \quad \sum_i \pi_i = 1\ . \end{aligned} \tag{4.2.26}$$

Mit der Festlegung $\pi_1 = 1, \pi_i = 0,\ i = 2, 3, \ldots, I$ kann der Start im Zustand S_1 erzwungen werden. Die generierten Zustände sind **nicht beobachtbar** – daher der Zusatz *"hidden"*.

Der *zweite Mechanismus* generiert eine Beobachtung o_i, wenn das Modell im Zustand S_i ist. Beobachtbar ist die Folge der *Beobachtungen* $\boldsymbol{o} = [o_i]$, die aus einer endlichen Menge von *Ausgabesymbolen* oder Beobachtungssymbolen

$$O = \{O_1, O_2, \ldots, O_L\} \tag{4.2.27}$$

gewählt wird. Ein Ausgabesymbol wird mit einer Wahrscheinlichkeit ausgewählt, die durch die Matrix der **Ausgabewahrscheinlichkeiten** definiert ist zu

$$\begin{aligned} \boldsymbol{B}^{I\times L} &= (b_{il})\ , \quad i = 1, \ldots, I\ , \quad l = 1, \ldots, L\ , \\ b_{il} &= P\left(O_l \text{ wird emittiert im Zustand } S_i \,|\, s_n = S_i\right) \\ &= P\left(o_n = O_l \,|\, s_n = S_i\right)\ , \quad 0 \leq b_{il}\ , \quad \sum_l b_{il} = 1\ . \end{aligned} \tag{4.2.28}$$

Damit werden Varianten der Aussprache von Wörtern und Fehler in der Erkennung einzelner Wortuntereinheiten (z. B. Laute) modelliert. Neben den oben angegebenen diskreten Ausgabewahrscheinlichkeiten können auch andere verwendet werden, z. B. eine Mischung aus Normalverteilungen.

Definition 4.6 *Ein* MARKOV-**Modell** *bzw. ein* **Hidden-**MARKOV-**Modell** *(*HMM*) ist definiert durch das Tripel*

$$\boxed{\text{HMM} = (\boldsymbol{\pi},\ \boldsymbol{A},\ \boldsymbol{B})} \tag{4.2.29}$$

der Anfangs–, Zustandsübergangs– und Ausgabewahrscheinlichkeiten.

Ein Wort w wird also wie folgt generiert. Mit der Wahrscheinlichkeit π_i wird das Zustandssymbol S_i als Anfangszustand s_1 gewählt. Von S_i erfolgt ein Zustandsübergang nach S_j mit der Wahrscheinlichkeit a_{ij}. Im Zustand S_i wird das Ausgabesymbol O_l mit der Wahrscheinlichkeit b_{il} emittiert. Die Schritte Zustandsübergang und Emission einer Ausgabe werden so oft wiederholt, bis eine Beobachtung der Länge T generiert wurde.

Je nach Einschränkung der Zustandsübergangswahrscheinlichkeiten unterscheidet man z. B. *ergodische* HMM, bei denen alle Zustandsübergänge möglich sind und *links–rechts* HMM, bei denen zeitlich rückwärts gerichtete Zustandsübergänge ausgeschlossen sind. Diese links–rechts–Modelle sind die in der Worterkennung verwendeten, da sie dem Fortschreiten in der Zeit entsprechen.

Bei den Hidden–MARKOV-Modellen wird die Struktur einmal festgelegt und nur die Parameter im Training berechnet; dafür können die Ausgabewahrscheinlichkeiten diskret oder kontinuierlich durch Mischungsverteilungen modelliert werden. Bei den *strukturierten* MARKOV-*Modellen*, für die auf die Literatur verwiesen wird, kann auch die Struktur im Training verändert werden, jedoch wird nur eine Normalverteilung verwendet.

Statistische Objektmodelle

An ein statistisches Objektmodell können unterschiedliche Anforderungen gestellt werden, insbesondere dass es *nur* Information über die Klasse und *nicht* über die Objektlage enthält, oder dass es neben der Information über die Objekt*klasse* auch Information über die Objekt*lage* in der Ebene oder im Raum enthält. Die klassenbedingte Verteilungsdichte $p(\boldsymbol{c}|\Omega_\kappa) = p(\boldsymbol{c}|\boldsymbol{a}_\kappa)$, in der der Parametervektor $\boldsymbol{a}_\kappa$ die *klassen*spezifischen Parameter enthält, muss daher im zweiten Fall um einen Parametervektor $\boldsymbol{\theta}$ erweitert werden, der die *lage*spezifischen Parameter enthält. Dieses sind im Raum die drei Translationen im Vektor $\boldsymbol{t}$ und die drei Rotationen in der Matrix $\boldsymbol{R}$. Das ergibt eine Dichte der Form $p(\boldsymbol{c}|\boldsymbol{a}_\kappa, \boldsymbol{\theta}) = p(\boldsymbol{c}|\boldsymbol{a}_\kappa, \boldsymbol{t}, \boldsymbol{R})$.

Ein globaler Merkmalsvektor $\boldsymbol{c}$, der aus einem Objekt extrahiert wird, ist bereits ein recht spezieller Ansatz. In Abschnitt 3.7.2 wurde der *globale* Merkmalsvektor $\boldsymbol{c}$ zu einer Menge *lokaler* Vektoren $\{(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\} = \{(j', k', \boldsymbol{c}_{j',k'})^\mathsf{T}\}$ verallgemeinert. Wenn man einfach den Grauwert als lokalen „Merkmalsvektor“ $\boldsymbol{c}_{\boldsymbol{x}}$ verwendet, besteht eine direkte Analogie zur Notation in (1.2.6), S. 13, bzw. zur Morphologie in (2.4.14), S. 111.

Damit ergibt sich als allgemeiner Ansatz für ein **statistisches Objektmodell** die Wahrscheinlichkeitsdichte, ein bestimmtes Bild $\boldsymbol{f}$, repräsentiert durch lokale Merkmale $\boldsymbol{c}_{\boldsymbol{x}}$, zu beobachten, wenn das Objekt der Klasse Ω_κ in der Lage $\boldsymbol{\theta} = (\boldsymbol{t}, \boldsymbol{R})$ vorliegt, zu

$$\begin{aligned} p(\boldsymbol{f}\,|\,\Omega_\kappa) &= p\left(\{(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\}\,|\,\boldsymbol{a}_\kappa,\, \boldsymbol{\theta}\right) \\ &= p\left(\{(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\}\,|\,\boldsymbol{a}_\kappa,\, \boldsymbol{t}, \boldsymbol{R}\right) \ . \end{aligned} \tag{4.2.30}$$

Lokalisation und Klassifikation erfordern dann das Lösen der Optimierungsprobleme

$$\boxed{\begin{aligned} \{\widehat{\boldsymbol{t}}_\kappa,\, \widehat{\boldsymbol{R}}_\kappa\} &= \underset{\boldsymbol{t},\boldsymbol{R}}{\operatorname{argmax}}\, p\left(\{(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\}\,|\,\boldsymbol{a}_\kappa,\, \boldsymbol{t}, \boldsymbol{R}\right) \ , \quad \kappa = 1, \ldots, k \ , \\ \Omega_\kappa &= \underset{\lambda}{\operatorname{argmax}}\, p(\Omega_\lambda\,|\,\{(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\}) \\ &= \underset{\lambda}{\operatorname{argmax}}\, p(\Omega_\lambda)\, p\left(\{(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\}\,|\,\boldsymbol{a}_\lambda,\, \widehat{\boldsymbol{t}}_\lambda,\, \widehat{\boldsymbol{R}}_\lambda\right) \ . \end{aligned}} \tag{4.2.31}$$

Das rechnerisch aufwendige Problem ist die Berechnung von Schätzwerten $\{\widehat{\boldsymbol{t}}_\kappa,\, \widehat{\boldsymbol{R}}_\kappa\}$ für die Lageparameter, da dieses *globale* Optimierungsprobleme sind. Die Bestimmung der optimalen Klasse Ω_κ ist dagegen unproblematisch.

Die obige Formulierung ist allerdings so allgemein, dass geeignete vereinfachende Spezialisierungen erfoderlich sind. Eine mögliche Spezialisierung ist die Annahme der statistischen Unabhängigkeit der lokalen Merkmale, eine zweite die Beschränkung auf ein statisches Bild zur Klassifikation, eine dritte die Faktorisierung der Dichte nach dem BAYES-Theorem über bedingte Dichten. Damit ergeben sich z. B. die Spezialisierungen

$$\begin{aligned} p(\boldsymbol{f}\,|\,\Omega_\kappa) &= p\left(\{(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\}\,|\,\boldsymbol{a}_\kappa,\, \boldsymbol{t}, \boldsymbol{R}\right) \\ &= \prod_{\boldsymbol{x}} p\left((\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}})^\mathsf{T}\,|\,\boldsymbol{a}_\kappa,\, \boldsymbol{t}, \boldsymbol{R}\right) && (4.2.32) \\ &= \prod_{\boldsymbol{x}} p(\boldsymbol{c}_{\boldsymbol{x}})\, p\left(\boldsymbol{x}\,|\,\boldsymbol{c}_{\boldsymbol{x}}, \boldsymbol{a}_\kappa,\, \boldsymbol{t}, \boldsymbol{R}\right) && (4.2.33) \\ &= \prod_{\boldsymbol{x}} p(\boldsymbol{x})\, p\left(\boldsymbol{c}_{\boldsymbol{x}}\,|\,\boldsymbol{x}, \boldsymbol{a}_\kappa,\, \boldsymbol{t}, \boldsymbol{R}\right) \ . && (4.2.34) \end{aligned}$$

Eine vierte Spezialisierung ist die konkrete Festlegung der Merkmale $\boldsymbol{c_x}$, die hier aber noch offen gelassen wird. Eine fünfte Spezialisierung ist die Berechnung lokaler Merkmale nicht in jedem Bildpunkt sondern nur in jedem m–ten sowie die Einführung einer Auflösungshierarchie.

Die zunächst rein formalen unterschiedlichen Faktorisierungen in (4.2.33) und (4.2.34) bedeuten unterschiedliche Ansätze für die Modellierung. Für ein einfaches Beispiel wird als Merkmalsvektor $\boldsymbol{c_x}$ der Grauwert f verwendet, der Koordinatenvektor $\boldsymbol{x}$ auf eine Variable x beschränkt und die Bedingung durch Objektklasse und –lage fortgelassen. Damit reduzieren sich die bedingten Dichten auf $p\left(\boldsymbol{x}\,|\,\boldsymbol{c_x}, \boldsymbol{a}_\kappa, \boldsymbol{t}, \boldsymbol{R}\right) \rightarrow p(x\,|\,f)$ und $p\left(\boldsymbol{c_x}\,|\,\boldsymbol{x}, \boldsymbol{a}_\kappa, \boldsymbol{t}, \boldsymbol{R}\right) \rightarrow p(f\,|\,x)$. Im Falle von (4.2.33) wird die Verteilungsdichte von x bedingt durch einen bestimmten Grauwert f angegeben. Da die Merkmale im Prinzip beliebig sind, kann man z. B. auch die Verteilungsdichte von x, allgemeiner von $(x, y, t)^\mathsf{T}$, bedingt durch einen Eckpunkt oder eine Kante angeben. Im Falle von (4.2.34) wird die Verteilungsdichte der Grauwerte f bedingt durch einen bestimmten Wert von x angegeben. Statt des Grauwertes f kann man z. B. auch die Verteilung von Waveletkoeffizienten $\boldsymbol{c}_{(x,y)}$ bedingt durch die Position (x, y) angeben.

Die Fülle der Möglichkeiten zur Definition stochastischer Objektmodelle ist damit angedeutet aber nicht ausgeschöpft.

4.2.2 Parameterschätzung

Die Konstruktion statistischer Modelle erfordert die Schätzung von Wahrscheinlichkeiten und Modellparametern, wie der letzte Abschnitt zeigte. Das Prinzip der Schätzung von Wahrscheinlichkeiten ist das Abzählen der Häufigkeit von Ereignissen, das bereits in Definition 2.3, S. 79, oder in (3.9.9), S. 249, angewendet wurde.

Maximum-likelihood- und BAYES-Schätzung

Zwei typische Parameterschätzverfahren sind die Maximum-likelihood- und die BAYES-Schätzung. Beide setzen eine Stichprobe ω von beobachteten Werten voraus, die dem zu modellierenden Problemkreis entstammen und deren Elemente statistisch unabhängig sein müssen. Wenn je Klasse eine repräsentative Stichprobe

$$\omega_\kappa = \left\{ {}^{\varrho}\boldsymbol{c}_\kappa \,|\, \varrho = 1, \ldots, N_\kappa,\ {}^{\varrho}\boldsymbol{c}_\kappa \in \mathbb{R}^n \right\} \tag{4.2.35}$$

gegeben ist und die Stichprobenelemente *statistisch unabhängig* sind, so ist die Wahrscheinlichkeit für die Beobachtung der Stichprobe

$$p(\omega_\kappa\,|\,\boldsymbol{a}_\kappa) = \prod_{\varrho=1}^{N_\kappa} p({}^{\varrho}\boldsymbol{c}_\kappa\,|\,\boldsymbol{a}_\kappa)\ , \quad \boldsymbol{a}_\kappa \in \mathbb{R}^m\ . \tag{4.2.36}$$

Die Maximum-likelihood-Schätzung berechnet denjenigen Schätzwert der Parameter einer Verteilungsdichte, der die Wahrscheinlichkeit der Beobachtung der Stichprobe maximiert.

Definition 4.7 *Der* **Maximum-likelihood-Schätzwert** *(MLS) des Parameters $\boldsymbol{a}_\kappa$ ist derjenige Wert $\widehat{\boldsymbol{a}}_\kappa$, der die Wahrscheinlichkeit der Stichprobe maximiert, d. h.*

$$\widehat{\boldsymbol{a}}_\kappa \quad = \quad \underset{\boldsymbol{a}_\kappa}{\operatorname{argmax}}\, p(\omega_\kappa\,|\,\boldsymbol{a}_\kappa)\ . \tag{4.2.37}$$

Statt $p(\omega_\kappa|\boldsymbol{a}_\kappa)$ kann in (4.2.37) auch eine monotone Funktion von $p(\cdot)$ verwendet werden. Wegen (4.2.36) ist der Logarithmus eine geeignete Funktion und ergibt

$$\widehat{\boldsymbol{a}}_\kappa = \underset{\boldsymbol{a}_\kappa}{\operatorname{argmax}} \sum_{\varrho=1}^{N_\kappa} \log\left[p({}^{\varrho}\boldsymbol{c}_\kappa|\boldsymbol{a}_\kappa)\right] \tag{4.2.38}$$

als äquivalente Formulierung des Maximum-likelihood-Schätzwertes für die Parameter $\boldsymbol{a}_\kappa$. Da damit das Produkt der Verteilungsdichten in eine Summe übergeführt wird, ist dieses die übliche Basis für konkrete Berechnungen.

Zur Berechnung eines BAYES-Schätzwerts wird der Parametervektor als Zufallsvariable mit der a priori Verteilungsdichte $p(\boldsymbol{a}_\kappa)$ aufgefasst. Die Wahl dieser Dichte ist nicht kritisch, jedoch darf der korrekte Parameterwert nicht ausgeschlossen werden. Nach Beobachtung der Stichprobe ω_κ hat der Parameter die a posteriori Verteilungsdichte

$$p(\boldsymbol{a}_\kappa|\omega_\kappa) = \frac{p(\boldsymbol{a}_\kappa)p(\omega_\kappa|\boldsymbol{a}_\kappa)}{p(\omega_\kappa)} \; . \tag{4.2.39}$$

Definition 4.8 *Der* BAYES*-Schätzwert oder* **Maximum-a-posteriori-Schätzwert** *(MAPS) maximiert die a posteriori Wahrscheinlichkeit des Parametervektors und ist gegeben durch*

$$\widehat{\boldsymbol{a}}_\kappa \;=\; \underset{\boldsymbol{a}_\kappa}{\operatorname{argmax}}\, p(\boldsymbol{a}_\kappa|\omega_\kappa) \; . \tag{4.2.40}$$

Zur Konvergenz dieser Schätzwerte wird auf die Literatur verwiesen. Es sei angemerkt, dass der MLS für große Werte N_κ erwartungstreu ist – d. h. der Erwartungswert des Schätzwertes ist gleich dem tatsächlichen Parameterwert – und dass die a posteriori Dichte für den MAPS unter recht allgemeinen Bedingungen gegen eine δ-Funktion strebt, die an der Stelle des richtigen Parameterwertes liegt; dafür ist es vor allem wichtig, dass, wie schon erwähnt, der richtige Parameterwert durch $p(\boldsymbol{a}_\kappa)$ nicht ausgeschlossen wird.

Grundsätzlich sind beide Schätzwerte „richtig“ im Sinne der Definition, trotzdem können sie verschieden sein. Allerdings werden beide bei sehr zuverlässigen Schätzungen sehr ähnlich, da wegen (4.2.58) für sehr große Werte von $p(\omega_\kappa|\boldsymbol{a}_\kappa)$ auch $p(\boldsymbol{a}_\kappa|\omega_\kappa)$ sehr groß wird. Darüberhinaus gibt es weitere Ansätze zur Berechnung von Schätzwerten, die auf anderen Kriterien beruhen. Dazu gehören die Minimierung des Fehlers bei der Klassifikation, die Maximierung der Transinformation oder die Maximierung der Entropie in Abschnitt 4.2.4. Auf *nichtparametrische Schätzungen* wird kurz in Abschnitt 4.2.6 eingegangen; Histogramme sind ein Beispiel für nichtparametrische Schätzungen.

Diskriminative Schätzung

Neben den beiden obigen Standardansätzen zur Definition von Schätzwerten gibt es weitere, von denen noch zwei vorgestellt werden, nämlich die diskriminative Schätzung und die Entropie Schätzung. Da bei der Klassifikation von Mustern oft die Minimierung der Fehlerrate angestrebt wird, was nach Satz 4.3, S. 315, durch Maximierung der a posteriori Wahrscheinlichkeit der gewählten Klasse erreicht wird, ist es sinnvoll, solche Schätzwerte zu berechnen, die die a posteriori Wahrscheinlichkeit der Klassen maximieren; diese sind in (4.1.34), S. 315, angegeben, wobei $p(\boldsymbol{c}|\Omega_\lambda) = p(\boldsymbol{c}|\boldsymbol{a}_\lambda)$ ist.

Definition 4.9 *Der* **diskriminative Schätzwert** *maximiert die a posteriori Wahrscheinlichkeit der Klasse und ist gegeben durch*

$$\widehat{\boldsymbol{a}}_\kappa = \underset{\boldsymbol{a}_\kappa}{\operatorname{argmax}} \sum_{\varrho=1}^{N_\kappa} \log\left[p(\Omega_\kappa|{}^\varrho\boldsymbol{c}_\kappa)\right] \ . \tag{4.2.41}$$

Die obige Schätzung wird auch als MMI–Schätzung ("maximum mutual information", maximale Transinformation) bezeichnet.

Maximum Entropie Schätzung

Die Maximierung der Entropie bedeutet, dass man einen Schätzwert sucht, der einerseits möglichst gut die Stichprobe repräsentiert, andererseits möglichst wenig Annahmen über diese impliziert. Auf diese Vorgehensweise wird in Abschnitt 4.2.4 genauer eingegangen.

Fehlende Information

Wir sind bisher von einer klassifizierten Stichprobe ausgegangen, d. h. von jedem Muster war seine Klassenzugehörigkeit bekannt. In der Objekterkennung geht man z. T. von „markierten" Stichproben aus, d. h. man erwartet, dass die *Korrespondenz* zwischen einem Merkmal des Objektmodells und einem Merkmal in einem Bild des Objekts bekannt ist. In der Spracherkennung ging man anfänglich von „fein" markierten Stichproben aus, d. h. für jedes Datenfenster von etwa 10 ms Dauer war bekannt, welcher Laut oder welche Lautkomponente gesprochen wurde. Offensichtlich erhöht das den Aufwand für die Stichprobensammlung signifikant. Es fragt sich also, ob man auf einige Information in der Stichprobe verzichten und trotzdem noch die für ein statistisches Modell erforderlichen Parameter schätzen kann. Theorie und Praxis zeigen, dass Parameterschätzung für viele Fälle fehlender Information möglich ist. Die empirische Basis geht auf frühe Ansätze zum *entscheidungsüberwachten Lernen* zurück, die theoretische Basis ist der **EM–Algorithmus**, dessen Prinzip in Abschnitt 1.6.4 erläutert wurde.

Informell besteht das Prinzip darin, die zu schätzenden Parameter (z. B. Mittelwert und Kovarianzmatrix von k Normalverteilungen) mit beliebigen Werten zu initialisieren, dann die fehlende Information zu schätzen (z. B. die Klassenzugehörigkeit von Merkmalsvektoren) und damit neue verbesserte Parameterwerte zu schätzen; dieser Prozess wird bis zur Konvergenz iteriert. Eine genauere Darstellung für die Schätzung der Parameter einer Mischung aus Normalverteilungen gibt Abschnitt 4.8.3.

Sparsame Schätzung

Die obigen Schätzverfahren liefern Parametervektoren mit voller, u. U. sehr großer, Dimension. Es kann sein, dass einige Komponenten des Parametervektors für die Güte von Klassifikationsergebnissen (oder von Regressionsergebnissen) unerheblich bzw. vernachlässigbar sind. Als **sparsamer Schätzwert** ("sparse estimate") wird ein solcher bezeichnet, der vernachlässigbare Komponenten eines Parametervektors unterdrückt, z. B. indem sie den Schätzwert Null erhalten.

Das Prinzip wird kurz am Beispiel des Regressionsproblems (vgl. S. 304) erläutert. Gesucht ist eine Regressionsfunktion

$$y = \boldsymbol{a}^\mathsf{T}\boldsymbol{\varphi}(\boldsymbol{f}) \ . \tag{4.2.42}$$

Dafür steht eine Stichprobe zur Verfügung, deren Beobachtungen durch additives, weisses, normalverteiltes Rauschen n_ϱ beeinflusst sind. Die Beobachtungen sind also

$$y_\varrho = \boldsymbol{a}^\mathsf{T}\boldsymbol{\varphi}({}^\varrho\boldsymbol{f}) + n_\varrho \ , \quad \varrho = 1, \ldots, N \ , \tag{4.2.43}$$

wobei die n_ϱ als statistisch unabhängig mit Streuung σ vorausgesetzt werden. Die Verteilung der Stichprobe von Beobachtungen ist dann normalverteilt mit

$$p(\boldsymbol{y}\,|\,\boldsymbol{a}) = \mathcal{N}(\boldsymbol{y}\,|\,\boldsymbol{H}\boldsymbol{a}, \sigma^2\boldsymbol{I}) \ . \tag{4.2.44}$$

Die Matrix $\boldsymbol{H}$ enthält die Beobachtungen in ihren mit $h_{ij} = \varphi_j({}^i\boldsymbol{f})$ definierten Elementen. Um einen MAP–Schätzwert zu berechnen, braucht man, wie oben erwähnt, eine a priori Verteilung für die zu schätzenden Parameter. Je nach Annahme über diese a priori Verteilung ergeben sich unterschiedliche Schätzwerte.

Eine mögliche und übliche Annahme besteht darin, für den Parametervektor $\boldsymbol{a}$ eine Normalverteilung mit Mittelwert Null und Kovarianzmatrix $\boldsymbol{A}$ vorzugeben, d. h. $p(\boldsymbol{a}) = \mathcal{N}(\boldsymbol{a}\,|\,\boldsymbol{0}, \boldsymbol{A})$. Es ist bekannt, dass die a posteriori Verteilung $p(\boldsymbol{a}\,|\,\boldsymbol{y})$ (s. (4.2.39)) wieder eine Normalverteilung ist mit dem Mittelwert, der auch der MAP–Schätzwert ist,

$$\widehat{\boldsymbol{a}} = \left(\sigma^2\boldsymbol{A}^{-1} + \boldsymbol{H}^\mathsf{T}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^\mathsf{T}\boldsymbol{y} \ . \tag{4.2.45}$$

Man sieht, dass dieser Schätzwert für $\boldsymbol{a}$ *nicht* sparsam ist. Wenn $\boldsymbol{A}$ die Form $\boldsymbol{A} = \beta\boldsymbol{I}\,,\ \beta \to \infty$, hat, geht der MAP–Schätzwert in den mittleren quadratischen Schätzwert $\widehat{\boldsymbol{a}} = \left(\boldsymbol{H}^\mathsf{T}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^\mathsf{T}\boldsymbol{y}$ über.

Ein sparsamer Schätzwert ergibt sich, wenn man statt der GAUSS- eine LAPLACE-Verteilung

$$p(\boldsymbol{a}) = \prod_{\nu=1}^{n}\frac{\alpha}{2}\exp[-\alpha|a_\nu|] = \left(\frac{\alpha}{2}\right)^n\exp[-\alpha||\boldsymbol{a}||_1] \tag{4.2.46}$$

für die a priori Verteilung des Schätzwertes vorgibt. Dabei wird mit $||\boldsymbol{a}||_r^r = \sum_\nu |a_\nu|^r$ die L^r Norm bezeichnet. In diesem Fall ist der Schätzwert nicht mehr linear in den Beobachtungen $\boldsymbol{y}$ wie in (4.2.45), sondern gegeben durch

$$\widehat{\boldsymbol{a}} = \underset{\boldsymbol{a}}{\operatorname{argmin}}\left\{||\boldsymbol{H}\boldsymbol{a} - \boldsymbol{y}||_2^2 + 2\alpha\sigma^2||\boldsymbol{a}||_1\right\} \ . \tag{4.2.47}$$

Dieses wird auch als **LASSO–Schätzwert** ("least absolute shrinkage and selection operator") bezeichnet. Die Sparsamkeit des Schätzwertes resultiert daraus, dass die L^1 Norm rascher wächst als die L^2 Norm, wenn mehr Vektorkomponenten von Null verschieden sind. Zum Beispiel ist $||(1,0)^\mathsf{T}||_2 = ||(1/\sqrt{2}, 1/\sqrt{2})^\mathsf{T}|| = 1$, jedoch $||(1,0)^\mathsf{T}||_1 = 1 < ||(1/\sqrt{2}, 1/\sqrt{2})^\mathsf{T}||_1 = \sqrt{2}$. Besonders klar wird die Sparsamkeit der Schätzwerte, wenn man eine orthogonale Beobachtungsmatrix $\boldsymbol{H}$ annimmt. In diesem Falle erhält man die geschlossene Lösung

$$\begin{aligned}
\widehat{a}_\nu &= \underset{a_\nu}{\operatorname{argmin}}\left\{a_\nu^2 - 2a_\nu(\boldsymbol{H}^\mathsf{T}\boldsymbol{y})_\nu + 2\alpha\sigma^2|a_\nu|\right\} \\
&= \begin{cases} \operatorname{sign}\left((\boldsymbol{H}^\mathsf{T}\boldsymbol{y})_\nu\right)\, q & : \quad q > 0 \\ 0 & : \quad q \le 0 \end{cases} \\
q &= \left(|(\boldsymbol{H}^\mathsf{T}\boldsymbol{y})_\nu| - \alpha\sigma^2\right) \ ,
\end{aligned} \tag{4.2.48}$$

wobei $(\boldsymbol{H}^\mathsf{T}\boldsymbol{y})_\nu$ die ν–te Komponente des Vektors $\boldsymbol{H}^\mathsf{T}\boldsymbol{y}$ und sign die Vorzeichenfunktion ist.

Für Einzelheiten wird auf die zitierte Literatur verwiesen. Die Vorgehensweise bei der Support Vektor Maschine in Abschnitt 4.3 liefert mit den Support Vektoren bereits sparsame Schätzungen, die in der Relevance Vector Maschine weiter verbessert werden. Auch die numerische Berechnung der Parametermatrix eines Polynomklassifikators in Abschnitt 4.4 mit dem GAUSS–JORDAN-Algorithmus und Auswahl der $m' < m$ wichtigsten Komponenten liefert im obigen Sinne sparsame Schätzwerte, wenn auch keine MAP–Schätzwerte.

Schätzung der Modellordnung

Die obigen Schätzverfahren lieferten einen Wert $\widehat{\boldsymbol{a}}_\kappa$ für einen unbekannten Parametervektor $\boldsymbol{a}_\kappa \in \mathbb{R}^m$ mit reellwertigen Komponenten. Die Zahl m der Komponenten von $\boldsymbol{a}_\kappa$ kann selbst ein unbekannter Parameter sein, dessen Wert ebenfalls zu schätzen ist. Diese Zahl m von Komponenten wird als die **Modellordnung** bezeichnet, auch wenn diese Bezeichnung u. U. wenig angemessen ist. Beispiele für Modellordnungen in diesem Sinne sind die Zahl der Komponenten in der linearen Vorhersage (s. (3.6.3), S. 209) oder die Zahl der Komponenten einer Mischungsverteilung (s. (4.2.15), S. 327). Im Folgenden wird der Klassenindex κ fortgelassen und bei Bedarf die Zahl der Komponenten des Parametervektors $\boldsymbol{a}$ explizit durch die Notation $\boldsymbol{a}^{(m)}$ angegeben.

Ausgangspunkt der Schätzung der Modellordnung ist die Wahrscheinlichkeit $p(\boldsymbol{c}|\boldsymbol{a})$ eines Merkmals– oder allgemeiner Beobachtungsvektors $\boldsymbol{c} \in \mathbb{R}^n$, die durch den Parametervektor $\boldsymbol{a}$ bedingt wird. Die Hypothese, dass der Parametervektor genau m Komponenten hat, sei $H_m,\ m \in \{1, 2, \ldots, m_{\max}\}$. Die Verteilungsdichte der Beobachtungen unter der Hypothese H_m wird mit $p(\boldsymbol{c}|H_m)$ bezeichnet, und es ist

$$p(\boldsymbol{c}|\boldsymbol{a}^{(m)}) = p(\boldsymbol{c}|H_m)\ . \tag{4.2.49}$$

Die Wahl der Modellordnung kann dann als Klassifikationsproblem aufgefasst werden, wobei die Hypothesen H_m den Klassen Ω_κ entsprechen. Mit Rückgriff auf Satz 4.3, S. 315 berechnet man

$$p(H_m|\boldsymbol{c}) = \frac{p(\boldsymbol{c}|H_m)p(H_m)}{p(\boldsymbol{c})}\ . \tag{4.2.50}$$

Nimmt man gleichwahrscheinliche Hypothesen H_m an, so ist die beste Modellordnung m^* gegeben durch

$$m^* = \underset{m\in\{1,\ldots,m_{\max}\}}{\operatorname{argmax}}\ p(\boldsymbol{c}|H_m)\ . \tag{4.2.51}$$

Das Problem bei diesem Ansatz liegt darin, dass die geforderte Verteilundsdichte i. Allg. nicht bekannt ist. Statt der tatsächlichen Dichte $p(\boldsymbol{c}|\boldsymbol{a}^{(m)})$ wird man also eine Approximation oder Schätzung $\widehat{p}(\boldsymbol{c}|\boldsymbol{a}^{(m)})$ verwenden. Mit dem KULLBACK-LEIBLER-Abstand

$$D(p;\widehat{p}) = \int_{\mathbb{R}^n} p \ln \frac{p}{\widehat{p}}\,\mathrm{d}\boldsymbol{c}\ =\ E_p\{\ln p\} - E_p\{\ln \widehat{p}\} \tag{4.2.52}$$

kann der Unterschied zwischen der tatsächlichen und der geschätzten Dichte gemessen werden. Er wird minimiert, wenn die relative KULLBACK-LEIBLER-Information

$$I(p;\widehat{p}) = E_p\{\ln \widehat{p}\} \tag{4.2.53}$$

maximiert wird. Da, wie gesagt, die tatsächliche Verteilungsdichte nicht bekannt ist, kann auch I nur geschätzt werden. Wir geben hier ohne weiteren Beweis das Ergebnis für zwei Näherungslösungen an, nämlich die auf dem AKAIKE-**Informationskriterium** (AIC) und die auf dem BAYES-**Informationskriterium** (BIC) beruhenden. Sie ergeben für die Modellordnung

$$m^*_{\mathrm{AIC}} = \underset{m\in\{1,\ldots,m_{\max}\}}{\operatorname{argmin}} \left(2m - 2\ln\left[p(\boldsymbol{c}|\widehat{\boldsymbol{a}}^{(m)})\right]\right) , \tag{4.2.54}$$

$$m^*_{\mathrm{BIC}} = \underset{m\in\{1,\ldots,m_{\max}\}}{\operatorname{argmin}} \left(m\ln n - 2\ln\left[p(\boldsymbol{c}|\widehat{\boldsymbol{a}}^{(m)})\right]\right) . \tag{4.2.55}$$

Für weitere Einzelheiten wird auf die Literatur in Abschnitt 4.11 verwiesen.

Eine wichtige Annahme besteht darin, dass unter den Dichten $\widehat{p}$ die richtige ist. Dieses wird i. Allg. nicht der Fall sein, jedoch zeigen experimentelle Ergebnisse, dass die obigen Ansätze trotzdem gute Ergebnisse erbringen.

4.2.3 Rekursive Schätzung

Die Parameter von statistischen Klassifikatoren lassen sich auch durch iterative Lernprozesse ermitteln. Dafür werden MLS und MAPS betrachtet. Die Berechnung eines MLS $\widehat{\boldsymbol{a}}_\kappa$ erfolgt zweckmäßigerweise über die Logarithmierung von (4.2.36)

$$l(\boldsymbol{a}_\kappa) = \ln p(\omega_\kappa|\boldsymbol{a}_\kappa) = \sum_{\varrho=1}^{N_\kappa} \ln\left[p({}^{\varrho}\boldsymbol{c}|\boldsymbol{a}_\kappa)\right] \tag{4.2.56}$$

sowie Nullsetzen der Ableitung von $l(\boldsymbol{a}_\kappa)$

$$\frac{\partial l(\boldsymbol{a}_\kappa)}{\partial \boldsymbol{a}_\kappa} = \sum_{\varrho=1}^{N_\kappa} \frac{\partial \ln\left[p({}^{\varrho}\boldsymbol{c}|\boldsymbol{a}_\kappa)\right]}{\partial \boldsymbol{a}_\kappa} = \boldsymbol{0} . \tag{4.2.57}$$

Daraus resultiert ein Gleichungssystem, dessen Lösung den ML–Schätzwert des Parametervektors ergibt.

Im Unterschied zum MLS geht man beim MAPS von der Vorstellung aus, dass der unbekannte Parameter $\boldsymbol{a}_\kappa$ eine Zufallsvariable mit einer bekannten *a priori Verteilungsdichte* $p(\boldsymbol{a}_\kappa)$ ist. Wenn man ω_κ beobachtet hat, wird Information über $\boldsymbol{a}_\kappa$ gewonnen, und die a priori Dichte geht in eine *a posteriori Verteilungsdichte* $p(\boldsymbol{a}_\kappa|\omega_\kappa)$ über, die sich mit fortlaufend verbesserter Schätzung immer mehr um den tatsächlichen Wert $\boldsymbol{a}$ konzentrieren sollte. Sie wird mit (4.2.39) aus

$$\begin{aligned} p(\boldsymbol{a}_\kappa|\omega_\kappa) &= \frac{p(\boldsymbol{a}_\kappa)p(\omega_\kappa|\boldsymbol{a}_\kappa)}{p(\omega_\kappa)} \\ &= \frac{p(\boldsymbol{a}_\kappa)p(\omega_\kappa|\boldsymbol{a}_\kappa)}{\int\limits_{\mathbb{R}_{\boldsymbol{a}}} p(\boldsymbol{a}_\kappa)p(\omega_\kappa|\boldsymbol{a}_\kappa)\,\mathrm{d}\boldsymbol{a}_\kappa} \end{aligned} \tag{4.2.58}$$

berechnet. Den MAPS $\widehat{\boldsymbol{a}}_\kappa$ von $\boldsymbol{a}_\kappa$ erhält man dann aus (4.2.40); er ist also der Wert, für den die a posteriori Dichte ihr Maximum annimmt. Für symmetrische, unimodale Dichten ist dieses der Mittelwert.

Rekursive MLS

Die wichtigste Aufgabe bei der Anwendung des MLS ist zunächst die Lösung von (4.2.57). Für einige Dichten sind solche Lösungen bekannt. Bereits in Abschnitt 4.2.1 wurden mit (4.2.8), (4.2.9) MLS für Mittelwert $\boldsymbol{\mu}_\kappa$ und Kovarianzmatrix $\boldsymbol{\Sigma}_\kappa$ einer *Normalverteilungsdichte* angegeben. Diese haben die Form $\boldsymbol{a}_\kappa = g(\omega_\kappa)$, sind also nicht direkt für die laufende Verbesserung der Schätzwerte durch Beobachtung neuer Muster geeignet. Es ist allerdings kein Problem, mit der in (4.4.33) angegebenen Methode die MLS auf die dort angegebene Form zu bringen. Zum Beispiel gilt für die mit N_κ Mustern ${}^\varrho\boldsymbol{c}_\kappa \in \omega_\kappa$, $\varrho = 1, \ldots, N_\kappa$ berechneten MLS von $\boldsymbol{\mu}_\kappa$ und $\boldsymbol{\Sigma}_\kappa$ die rekursive Beziehung

$$\boxed{\begin{aligned}
\widehat{\boldsymbol{\mu}}_{\kappa N_\kappa} &= \frac{(N_\kappa - 1)}{N_\kappa}\widehat{\boldsymbol{\mu}}_{\kappa,N_\kappa-1} + \frac{1}{N_\kappa}{}^{N_\kappa}\boldsymbol{c}_\kappa \;, \\
\widehat{\boldsymbol{\Sigma}}_{\kappa N_\kappa} &= \frac{(N_\kappa - 1)}{N_\kappa}\widehat{\boldsymbol{\Sigma}}_{\kappa,N_\kappa-1} + \frac{(N_\kappa - 1)}{N_\kappa^2}({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa,N_\kappa-1})({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa,N_\kappa-1})^\mathsf{T} \\
&= \frac{(N_\kappa - 1)}{N_\kappa}\widehat{\boldsymbol{\Sigma}}_{\kappa,N_\kappa-1} + \frac{1}{(N_\kappa - 1)}({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa N_\kappa})({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa N_\kappa})^\mathsf{T} \;.
\end{aligned}} \tag{4.2.59}$$

Die Initialisierung erfolgt mit $\widehat{\boldsymbol{\mu}}_{\kappa 0} = \mathbf{0}$ und $\widehat{\boldsymbol{\Sigma}}_{\kappa 0} = \mathbf{0}$. Damit ist eine laufende Verbesserung der Schätzwerte möglich.

Allerdings braucht man zur Klassifikation in (4.2.116) nicht $\widehat{\boldsymbol{\Sigma}}_{\kappa N_\kappa}$ sondern $\widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa N_\kappa}$. Eine rekursive Berechnung von $\widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa N_\kappa}$ ist ebenfalls möglich. Sind nämlich $\boldsymbol{A}$ und $\boldsymbol{B}$ reguläre Matrizen, $\boldsymbol{x}$ ein Spaltenvektor, a eine reelle Zahl, $\boldsymbol{B}^{-1}$ die Inverse von $\boldsymbol{B}$ und gilt

$$\boldsymbol{A} = \boldsymbol{B} + a\boldsymbol{x}\boldsymbol{x}^\mathsf{T} \;, \tag{4.2.60}$$

dann ergibt sich die Inverse $\boldsymbol{A}^{-1}$ aus

$$\boldsymbol{A}^{-1} = \boldsymbol{B}^{-1} - \frac{a}{1 + a\boldsymbol{x}^\mathsf{T}\boldsymbol{B}^{-1}\boldsymbol{x}}\,\boldsymbol{B}^{-1}\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\boldsymbol{B}^{-1} \;. \tag{4.2.61}$$

Da die Berechnung von $\widehat{\boldsymbol{\Sigma}}_{\kappa N_\kappa}$ in (4.2.59) die Form von (4.2.60) hat, lässt sich $\widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa N_\kappa}$ durch $\widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa,N_\kappa-1}$ ausdrücken. Einsetzen ergibt

$$\boxed{\begin{aligned}
\widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa N_\kappa} &= \frac{N_\kappa}{N_\kappa - 1}\widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa,N_\kappa-1} - \frac{\boldsymbol{y}\boldsymbol{y}^\mathsf{T}}{(N_\kappa - 1)\left(1 + \dfrac{({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa N_\kappa})^\mathsf{T}\boldsymbol{y}}{(N_\kappa - 1)}\right)} \;, \\
\boldsymbol{y} &= \widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa,N_\kappa-1}\left({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa N_\kappa}\right) \;.
\end{aligned}} \tag{4.2.62}$$

Damit ist an einem Beispiel die rekursive Berechnung von MLS gezeigt.

Rekursive MAPS

Bei der Verwendung von MAPS ist zunächst die a posteriori Dichte in (4.2.58) zu berechnen, wobei hier eine rekursive Form angegeben wird. Wie üblich verwenden wir die Stichprobe $\omega_\kappa =$

$\{{}^{\varrho}\boldsymbol{c}_\kappa \,|\, \varrho = 1, \ldots, N_\kappa\}$ sowie zur Abkürzung eine Stichprobe $\omega'_\kappa = \{{}^{\varrho}\boldsymbol{c}_\kappa \,|\, \varrho = 1, \ldots, N_\kappa - 1\}$. Aus (4.2.58) ergibt sich dann

$$\begin{aligned} p(\boldsymbol{a}_\kappa \,|\, \omega_\kappa) &= \frac{p(\boldsymbol{a}_\kappa) p(\omega_\kappa \,|\, \boldsymbol{a}_\kappa)}{\int\limits_{\mathbb{R}_{\boldsymbol{a}}} p(\boldsymbol{a}_\kappa) p(\omega_\kappa \,|\, \boldsymbol{a}_\kappa)\, \mathrm{d}\boldsymbol{a}_\kappa} \\ &= \frac{p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa \,|\, \boldsymbol{a}_\kappa)\, p(\omega'_\kappa \,|\, \boldsymbol{a}_\kappa)}{\int\limits_{\mathbb{R}_{\boldsymbol{a}}} p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa \,|\, \boldsymbol{a}_\kappa)\, p(\omega'_\kappa \,|\, \boldsymbol{a}_\kappa)\, \mathrm{d}\boldsymbol{a}_\kappa} \\ &= \frac{p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa \,|\, \boldsymbol{a}_\kappa)\, \dfrac{p(\boldsymbol{a}_\kappa \,|\, \omega'_\kappa) p(\omega'_\kappa)}{p(\boldsymbol{a}_\kappa)}}{\int\limits_{\mathbb{R}_{\boldsymbol{a}}} p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa \,|\, \boldsymbol{a}_\kappa)\, \dfrac{p(\boldsymbol{a}_\kappa \,|\, \omega'_\kappa) p(\omega'_\kappa)}{p(\boldsymbol{a}_\kappa)}\, \mathrm{d}\boldsymbol{a}_\kappa} \\ &= \frac{p({}^{N_\kappa}\boldsymbol{c}_\kappa \,|\, \boldsymbol{a}_\kappa)\, p(\boldsymbol{a}_\kappa \,|\, {}^{1}\boldsymbol{c}_\kappa, \ldots, {}^{N_\kappa - 1}\boldsymbol{c}_\kappa)}{\int\limits_{\mathbb{R}_{\boldsymbol{a}}} p({}^{N_\kappa}\boldsymbol{c}_\kappa \,|\, \boldsymbol{a}_\kappa) p(\boldsymbol{a}_\kappa \,|\, {}^{1}\boldsymbol{c}_\kappa, \ldots, {}^{N_\kappa - 1}\boldsymbol{c}_\kappa)\, \mathrm{d}\boldsymbol{a}_\kappa} \,. \end{aligned} \tag{4.2.63}$$

Dieses ist die Basis der rekursiven MAPS bzw. des BAYES-*Lernens*. Dabei treten i. Allg. zwei Probleme auf. Zum einen werden zur Berechnung eines verbesserten Schätzwertes *alle* vorher beobachteten Muster gebraucht, zum anderen kann $p(\boldsymbol{a}_\kappa \,|\, {}^{1}\boldsymbol{c}_\kappa, \ldots, {}^{N_\kappa - 1}\boldsymbol{c}_\kappa)$ eine *andere* Funktion sein als $p(\boldsymbol{a}_\kappa \,|\, {}^{1}\boldsymbol{c}_\kappa, \ldots, {}^{N_\kappa}\boldsymbol{c}_\kappa)$. Es gibt aber Spezialfälle, in denen diese allgemeine Schätzgleichung (4.2.63) sich auf eine rekursiv auswertbare Form reduziert. Die Voraussetzungen dafür sind, dass es eine einfache hinreichende Statistik $\boldsymbol{s}$ zur Schätzung von $\boldsymbol{a}_\kappa$ gibt und dass es eine selbstreproduzierende Verteilungsdichte $p(\boldsymbol{a}_\kappa)$ gibt; beide Begriffe werden noch näher erläutert.

Es sei

$$\boldsymbol{s}(\omega_\kappa) = (s_1(\omega_\kappa), \ldots, s_l(\omega_\kappa))^{\mathsf{T}} \tag{4.2.64}$$

eine Statistik, also eine Funktion von ω_κ. Eine hinreichende Statistik $\boldsymbol{s}$ enthält alle Information, die zum Schätzen von $\boldsymbol{a}_\kappa$ notwendig ist. Definitionsgemäß wird $\boldsymbol{s}$ als **hinreichende Statistik** bezeichnet, wenn

$$\boxed{p(\omega_\kappa \,|\, \boldsymbol{a}_\kappa, \boldsymbol{s}) = p(\omega_\kappa \,|\, \boldsymbol{s})} \tag{4.2.65}$$

ist, d. h. wenn $p(\omega_\kappa \,|\, \boldsymbol{a}_\kappa, \boldsymbol{s})$ *unabhängig* von $\boldsymbol{a}_\kappa$ ist. Damit ergibt sich

$$p(\boldsymbol{a}_\kappa \,|\, \boldsymbol{s}, \omega_\kappa) = \frac{p(\omega_\kappa \,|\, \boldsymbol{a}_\kappa, \boldsymbol{s}) p(\boldsymbol{a}_\kappa \,|\, \boldsymbol{s})}{p(\omega_\kappa \,|\, \boldsymbol{s})} = p(\boldsymbol{a}_\kappa \,|\, \boldsymbol{s}) \,, \tag{4.2.66}$$

d. h. die a posteriori Wahrscheinlichkeit von $\boldsymbol{a}_\kappa$ hängt nur von der hinreichenden Statistik $\boldsymbol{s}$ ab. Die wichtigste Aussage enthält

Satz 4.7 *(***Faktorisierungstheorem***) Eine notwendige und hinreichende Bedingung dafür, dass $\boldsymbol{s}(\omega_\kappa)$ eine hinreichende Statistik für $\boldsymbol{a}_\kappa$ ist, besteht darin, dass sich die Dichte $p(\omega_\kappa \,|\, \boldsymbol{a}_\kappa)$ faktorisieren lässt in*

$$p(\omega_\kappa \,|\, \boldsymbol{a}_\kappa) = g(\boldsymbol{s}(\omega_\kappa), \boldsymbol{a}_\kappa) h(\omega_\kappa) \,. \tag{4.2.67}$$

Beweis*: s. z. B. [Duda und Hart, 1972b], Sect. 3.6.*

Es ist bekannt, dass es eine hinreichende Statistik für die n-dimensionale Normalverteilungsdichte gibt (nämlich Mittelwertsvektor und Korrelationsmatrix) und dass es schon für die bewichtete Summe zweier eindimensionaler Normalverteilungen keine hinreichende Statistik gibt.

Setzt man (4.2.67) in (4.2.63) ein, ergibt sich

$$\begin{aligned} p(\boldsymbol{a}_\kappa|\omega_\kappa) &= \frac{p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa|\,\boldsymbol{a}_\kappa)\, g(\boldsymbol{s},\boldsymbol{a}_\kappa)\, h(\omega'_\kappa)}{\int\limits_{\mathbb{R}_{\boldsymbol{a}}} p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa|\,\boldsymbol{a}_\kappa)\, g(\boldsymbol{s},\boldsymbol{a}_\kappa)\, h(\omega'_\kappa)\,\mathrm{d}\boldsymbol{a}_\kappa} \\ &= \frac{p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa|\,\boldsymbol{a}_\kappa)\, g(\boldsymbol{s},\boldsymbol{a}_\kappa)}{\int\limits_{\mathbb{R}_{\boldsymbol{a}}} p(\boldsymbol{a}_\kappa)\, p({}^{N_\kappa}\boldsymbol{c}_\kappa|\,\boldsymbol{a}_\kappa)\, g(\boldsymbol{s},\boldsymbol{a}_\kappa)\,\mathrm{d}\boldsymbol{a}_\kappa}\,. \end{aligned} \tag{4.2.68}$$

Die vorher beobachtete Stichprobe $\omega' = \{{}^{\varrho}\boldsymbol{c}_\kappa|\,\varrho = 1,\ldots,N_\kappa - 1\}$ wird also in (4.2.68) nicht mehr gebraucht, sondern nur die hinreichende Statistik $\boldsymbol{s}$, deren Dimension l *unabhängig* vom Stichprobenumfang N_κ ist.

Eine a priori Verteilungsdichte $p(\boldsymbol{a}_\kappa)$ wird als **selbstreproduzierende Verteilungsdichte** bezeichnet, wenn

$$p(\boldsymbol{a}_\kappa|\,{}^1\boldsymbol{c}_\kappa) = \frac{p({}^1\boldsymbol{c}_\kappa|\,\boldsymbol{a}_\kappa)p(\boldsymbol{a}_\kappa)}{\int p({}^1\boldsymbol{c}_\kappa|\,\boldsymbol{a}_\kappa)p(\boldsymbol{a}_\kappa)\,\mathrm{d}\boldsymbol{a}_\kappa} \tag{4.2.69}$$

eine Funktion aus der gleichen parametrischen Familie wie $p(\boldsymbol{a}_\kappa)$ ist. Nach Beobachtung des ersten Musters ist (4.2.69) die a posteriori Dichte von $\boldsymbol{a}_\kappa$ gemäß (4.2.63). Damit ist auch für $N_\kappa > 1$ die Dichte $p(\boldsymbol{a}_\kappa|\omega_\kappa)$ eine Funktion aus der *gleichen* parametrischen Familie wie $p(\boldsymbol{a}_\kappa)$, und das Integral im Nenner von (4.2.63) muss *nur einmal* berechnet werden.

Die rekursive Berechnung von MAPS der Parameter $\boldsymbol{\mu}_\kappa$, $\boldsymbol{\Sigma}_\kappa$ einer Normalverteilungsdichte ist in der Literatur ausführlich behandelt. Hier werden nur die wichtigsten Ergebnisse der zum Teil längeren Rechnungen angegeben. Es ist bekannt, dass der Schätzwert für $\boldsymbol{\mu}_\kappa$ eine Normalverteilung hat, dass der Schätzwert für $\boldsymbol{L}_\kappa = \boldsymbol{\Sigma}_\kappa^{-1}$ eine WISHART-*Verteilungsdichte* hat und dass beide statistisch unabhängig sind. Es ist daher naheliegend, als a priori Dichte

$$p(\boldsymbol{a}_\kappa) = p(\boldsymbol{\mu}_\kappa, \boldsymbol{L}_\kappa) = \mathcal{N}(\boldsymbol{\mu}_\kappa|\,\boldsymbol{\mu}_0, \boldsymbol{\Phi}_0)\,\mathcal{W}(\boldsymbol{L}_\kappa|\,\nu_0, \boldsymbol{K}_0) \tag{4.2.70}$$

anzunehmen. Dabei ist $\mathcal{N}(\boldsymbol{\mu}_\kappa|\,\boldsymbol{\mu}_0, \boldsymbol{\Phi}_0)$ eine Normalverteilungsdichte (4.2.3), S. 324, mit dem Mittelwert $\boldsymbol{\mu}_0$ und der Kovarianzmatrix $\boldsymbol{\Phi}_0$ in (4.2.72), und die WISHART-Verteilung ist

$$\mathcal{W}(\boldsymbol{L}_\kappa|\,\nu_0, \boldsymbol{K}_0) = \alpha_{n,\nu_0}\left|\frac{\nu_0}{2}\boldsymbol{K}_{\mathbf{0}}\right|^{(\nu_0-1)/2}|\boldsymbol{L}_\kappa|^{(\nu_0-n-2)/2}\exp\left[-\frac{1}{2}\mathrm{Sp}\,(\nu_0\boldsymbol{L}_\kappa\boldsymbol{K}_0)\right]\;; \tag{4.2.71}$$

sie ist definiert in dem Bereich des $n(n+1)/2$–dimensionalen Raumes, in dem die $n \times n$ Matrix $\boldsymbol{L}_\kappa$ positiv definit ist, und sonst Null. Die Matrix $\boldsymbol{K}_0$ ist ein Anfangswert für $\boldsymbol{\Sigma}_\kappa$ und die Konstante $\nu_0 > n$ ein Maß für die Konzentration der Dichte um $\boldsymbol{K}_0^{-1}$, d. h. dafür wie zuverlässig der Anfangswert $\boldsymbol{K}_0$ für $\boldsymbol{\Sigma}_\kappa$ ist. Die Konstante α_{n,ν_0} normiert das Integral auf Eins. Entsprechend enthält $\boldsymbol{\mu}_0$ eine Annahme über $\boldsymbol{\mu}_\kappa$ und $\boldsymbol{\Phi}_0$ eine Annahme über die Konzentration von $\boldsymbol{\mu}_0$ um $\boldsymbol{\mu}_\kappa$. Zur Vereinfachung wird

$$\boldsymbol{\Phi}_0 = \frac{1}{\beta_0}\boldsymbol{\Sigma}_\kappa = \frac{1}{(\beta_0\boldsymbol{L}_\kappa)} \tag{4.2.72}$$

gesetzt. Aus der zitierten Literatur geht hervor, dass die Dichte $p(\boldsymbol{a}_\kappa)$ in (4.2.70) *selbstreproduzierend* ist. Man erhält die MAPS (bzw. die BAYES-Schätzwerte) nach Beobachtung von N_κ Mustern ${}^{\varrho}\boldsymbol{c}_\kappa \in \omega_\kappa$ aus denen nach der Beobachtung von $N_\kappa - 1$ Mustern zu

$$\boxed{\begin{aligned}\widehat{\boldsymbol{\mu}}_{\kappa N_\kappa} &= \frac{(\beta_0 + N_\kappa - 1)\widehat{\boldsymbol{\mu}}_{\kappa,N_\kappa-1} + {}^{N_\kappa}\boldsymbol{c}_\kappa}{(\beta_0 + N_\kappa)}\,, \\ \widehat{\boldsymbol{\Sigma}}_{\kappa N_\kappa} &= \frac{\nu_0 + N_\kappa - 1}{\nu_0 + N_\kappa}\,\widehat{\boldsymbol{\Sigma}}_{\kappa,N_\kappa-1} + \\ &\quad \frac{\beta_0 + N_\kappa - 1}{(\nu_0 + N_\kappa)(\beta_0 + N_\kappa)}\left({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa,N_\kappa-1}\right)\left({}^{N_\kappa}\boldsymbol{c}_\kappa - \widehat{\boldsymbol{\mu}}_{\kappa,N_\kappa-1}\right)^{\mathsf{T}}\,.\end{aligned}} \tag{4.2.73}$$

Dabei ist $\widehat{\boldsymbol{\mu}}_{\kappa 0} = \boldsymbol{\mu}_0$ und $\widehat{\boldsymbol{\Sigma}}_{\kappa 0} = \boldsymbol{K}_0$. Die rekursive Berechnung von $\widehat{\boldsymbol{\Sigma}}^{-1}_{\kappa N_\kappa}$ ist mit (4.2.61) ebenfalls möglich.

Ein Vergleich der MLS (4.2.59) mit den MAPS (4.2.73) zeigt, dass beide sich nur in der Verwendung von Anfangswerten für die Schätzung unterscheiden. Setzt man $\nu_0 = \beta_0 = 0, \boldsymbol{\mu}_{\kappa 0} = \mathbf{0}, \widehat{\boldsymbol{\Sigma}}_{\kappa 0} = \mathbf{0}$, geht (4.2.73) in (4.2.59) über. Das Lernen von Parametern statistischer Klassifikatoren durch laufende Verbesserung der Parameter mit neu beobachteten Mustern ist also problemlos, wenn die Muster *bekannte Klassenzugehörigkeit* haben (überwachtes Lernen) und wenn die Merkmale *klassenweise normalverteilt* sind. Es kann zweckmäßig sein, mit festen Gewichten zu arbeiten. Zum Beispiel würde man dann den Schätzwert für $\boldsymbol{\mu}_\kappa$ in (4.2.59) ersetzen durch

$$\boxed{\widehat{\boldsymbol{\mu}}_{\kappa N_\kappa} = (1-\beta)\widehat{\boldsymbol{\mu}}_{\kappa,N_\kappa-1} + \beta^{N_\kappa}\boldsymbol{c}_\kappa\,, \quad 0 < \beta \leq 1\,.} \tag{4.2.74}$$

Natürlich ist das *kein* MLS , und die Konvergenz ist experimentell zu sichern durch geeignete Wahl von β. Der Vorteil einer festen Gewichtung besteht darin, dass eine relativ rasche *Adaptation* an neue Beobachtungen erfolgt, wobei die Geschwindigkeit der Adaptation von der Wahl von β abhängt.

In Abschnitt 4.2.1 wurde erwähnt, dass mit den Schätzwerten (4.2.8), (4.2.9) für $\boldsymbol{\mu}_\kappa, \boldsymbol{K}_\kappa$ so gerechnet wird, als seien es die richtigen Werte. Dieses ist natürlich nur eine näherungsweise Betrachtung. Genaugenommen ist die Verteilungsdichte $p(\boldsymbol{c}|\Omega_\kappa) = p(\boldsymbol{c}|\boldsymbol{a}_\kappa)$ der Merkmalsvektoren zu ersetzen durch $p(\boldsymbol{c}|\Omega_\kappa, \omega_\kappa)$, um den Einfluss der endlichen Stichprobe zu berücksichtigen. Dieser besteht darin, dass man nicht die tatsächlichen Parametwerte $\boldsymbol{a}_\kappa$ zur Verfügung hat sondern Schätzwerte $\widehat{\boldsymbol{a}}_\kappa$ mit der Verteilungsdichte $p(\boldsymbol{a}_\kappa|\omega_\kappa)$. Die gesuchte Verteilungsdichte der Merkmalsvektoren ist damit

$$p(\boldsymbol{c}|\Omega_\kappa, \omega_\kappa) = \int_{\mathbb{R}_{\boldsymbol{a}_\kappa}} p(\boldsymbol{c}|\boldsymbol{a}_\kappa)\, p(\boldsymbol{a}_\kappa|\omega_\kappa)\, \mathrm{d}\boldsymbol{a}_\kappa\,. \tag{4.2.75}$$

Setzt man für die Verteilungsdichte der Merkmalsvektoren eine GAUSS-Verteilung an und für die der Parameter (4.2.70), so ist die Integration in (4.2.75) ausführbar. Als wesentliche Ergebnisse sind festzuhalten, dass die resultierende Verteilungsdichte *keine* GAUSS-Verteilung ist, dass diese aber für großen Stichprobenumfang gegen eine GAUSS-Verteilung strebt und dass dieses praktisch schon für relativ kleinen Stichprobenumfang geschieht. In diesem Sinne ist es gerechtfertigt, mit den Schätzwerten so zu rechnen, als seien es die richtigen. Es bleibt dagegen das Problem, dass die Annahme klassenweise normalverteilter Merkmale i. Allg. nur eine Näherung an die tatsächlichen Verhältnisse ist. Bereits in Abschnitt 4.2.1 wurde darauf

hingewiesen, dass mit einer Mischung von Normalverteilungen gemäß (4.2.15) praktisch beliebige Verteilungsdichten approximierbar sind. Die Schätzung von deren Parametern wird in Abschnitt 4.8.2 behandelt.

4.2.4 Modelle mit maximaler Entropie

Motivation

Wir betrachten ein Zufallsexperiment mit K möglichen *Ereignissen*, das N mal ausgeführt wird, sodass K^N *Ergebnisse* möglich sind; d. h. ein Ergebnis (von N Ausführungen) besteht aus N Ereignissen und deren Ordnung. Jedes Ergebnis liefert absolute Häufigkeiten N_i und relative Häufigkeiten $\widehat{p}_i = N_i/N$, $i = 1, \ldots, K$ der beobachteten Ereignisse. Diese haben die *Entropie*

$$H(\widehat{p}_1, \ldots, \widehat{p}_K) = -\sum_{i=1}^{K} \widehat{p}_i \, \ln \widehat{p}_i \; . \tag{4.2.76}$$

Beispiele sind ein Würfel mit K Flächen, der N mal geworfen wird und die i–te Fläche N_i mal zeigt, oder ein Bild mit Gesamthelligkeit N, die auf K Bildpunkte so verteilt ist, dass der i-te Bildpunkt den Anteil $\widehat{p}_i = N_i/N$ der Gesamthelligkeit erhält.

Das Ergebnis eines Zufallsexperiments mit absoluten Häufigkeiten N_i kann auf viele Arten realisiert werden, indem die Ordnung der Ereignisse permutiert wird. Die Vielfachheit V ist

$$V = \frac{N!}{N_1! \, N_2! \, \ldots \, N_K!} \; . \tag{4.2.77}$$

Mit der STIRLING-Formel erhält man für sehr große Werte von N

$$\frac{1}{N} \ln V \approx -\sum_i \frac{N_i}{N} \ln \frac{N_i}{N} \; , \tag{4.2.78}$$

d. h. die Entropie (4.2.76) nach SHANNON. Die Bevorzugung von Verteilungen mit hoher Entropie bedeutet die Bevorzugung von Verteilungen mit größerer Vielfachheit; sie können („von der Natur“) auf viele verschiedene Arten realisiert werden.

Weiterhin seien $m < K$ linear unabhängige *Nebenbedingungen* der Form

$$\sum_{i=1}^{K} a_{ij} \widehat{p}_i = d_j \; , \quad 1 \le j \le m \tag{4.2.79}$$

gegeben. Diese bedeuten, dass m Messungen oder *Beobachtungen* gemacht wurden, wobei die Matrix $\boldsymbol{A} = [a_{ij}]$ deren Art bestimmt und der Vektor $\boldsymbol{d} = (d_1, \ldots, d_m)^\mathsf{T}$ die gemessenen Daten enthält. Es sei M die Menge aller Ergebnisse des Zufallsexperiments, die kompatibel mit den Nebenbedingungen (4.2.79) ist. Die Daten $\boldsymbol{d}$ erlauben nur den Schluss, dass das aktuelle Ergebnis aus M ist, bestimmen aber nicht die relativen Häufigkeiten $\widehat{p}_i$.

Ein bestimmter Anteil b der Ergebnisse aus M wird eine Entropie im Bereich

$$H_{\max} - \Delta H \le H(\widehat{p}_1, \ldots, \widehat{p}_K) \le H_{\max} \tag{4.2.80}$$

ergeben. Der Zusammenhang zwischen b und ΔH, d. h. die *Konzentration* der Entropie um die obere Grenze ist durch den folgenden Satz gegeben.

Satz 4.8 *(***Konzentration der Entropie***) Die Größe $2N\Delta H$ ist asymptotisch χ^2–verteilt mit $r = K - m - 1$ Freiheitsgraden, und zwar* unabhängig *von der Art der Nebenbedingungen. Die χ^2–Verteilung mit r Freiheitsgraden auf dem $100s\%$ Signifikanzniveau sei $\chi^2_r(s)$. Damit gilt*

$$2N\Delta H = \chi^2_r(1-b)\ . \tag{4.2.81}$$

Beweis*: s. z. B. [Jaynes, 1982].*

Die Unabhängigkeit von den Nebenbedingungen besagt, dass für *alle* Zufallsexperimente mit r Freiheitsgraden z. B. auf dem 95% Signifikanzniveau gilt

$$H_{\max} - \frac{\chi^2_r(0,05)}{2N} \leq H \leq H_{\max}\ . \tag{4.2.82}$$

Der Wert von $H_{\max}$ hängt allerdings vom Experiment und den Nebenbedingungen ab. Im Allgemeinen folgt aus dem Satz, dass die Breite des Intervalls (4.2.80) mit $1/N$ abnimmt.

Als *Beispiel* wird ein Experiment mit $K = 6$ (Würfel) und $N = 1000$ betrachtet. Es gebe keine Nebenbedingungen (ausser $\sum \widehat{p}_i = 1$). Es ist bekannt, dass die maximale Entropie $H_{\max} = \ln 6 = 1,792$ von der Gleichverteilung der Ereignisse erreicht wird. Eine χ^2–Verteilung mit $r = K - m - 1 = 6 - 1 = 5$ Freiheitsgraden auf dem Signifikanzniveau 95% hat den Wert $\chi^2_5(0,05) = 11,07$. Der Konzentrationssatz sagt, dass $2N\Delta H = 11,07$ ist und damit 95% *aller möglichen* Ergebnisse eine Entropie im Bereich

$$H_{\max} - \Delta H = 1,786 \leq H \leq 1,792 \tag{4.2.83}$$

haben, oder anders gesagt, die weit überwiegende Mehrheit der Ergebnisse wird eine Verteilung haben, die sehr nahe an der Gleichverteilung liegt. Man braucht also nicht die „Intuition“ zu bemühen, um auf die Gleichverteilung zu schließen, sondern diese folgt aus einem Häufigkeitsargument – man wird das annehmen, was aus kombinatorischen Gründen am häufigsten zu erwarten ist.

Betrachten wir nun das gleiche Experiment wie im letzten Absatz und nehmen zusätzlich an, dass die Nebenbedingung

$$\sum_{i=1}^{6} i\,\widehat{p}_i = 4,5 \tag{4.2.84}$$

gegeben ist, d. h. die mittlere gewürfelte Augenzahl ist nicht mehr $3,5$ wie bei einem echten Würfel. Was kann man nun über die relativen Häufigkeiten $\widehat{p}_i$ sagen? Die Intuition wird hier nicht hilfreich sein. Man bestimmt wieder die relativen Häufigkeiten, die die Entropie maximieren mit dem MAXENT–Algorithmus (s. u.) und findet

$$(\widehat{p}_1, \ldots, \widehat{p}_6) = (0,0543,\ 0,0788,\ 0,1142,\ 0,1654,\ 0,2398,\ 0,3475) \tag{4.2.85}$$

sowie $H_{\max} = 1,614$. Eine Tabelle ergibt $2N\Delta H = \chi^2_4(0,05) = 9,48$, sodass die Entropie im Intervall $1,609 \leq H \leq 1,614$ liegt. Auch hier hat also die weit überwiegende Mehrheit der Ergebnisse eine Entropie, die recht nahe um die maximale konzentriert ist, und damit ist es sehr selten, dass die relativen Häufigkeiten andere sind als in (4.2.85).

Stochastische Modelle mit maximaler Entropie ordnen Ereignissen also die relativen Häufigkeiten zu, die zur *weit überwiegenden* Mehrheit der Ergebnisse, die kompatibel mit den gemessenen Daten (Nebenbedingungen) sind, führen. Anders gesagt vermeidet man mit dieser

Vorgehensweise, dass man von den Daten auf ein Modell schließt, das irgendeine seltene Anomalie repräsentiert, für die es in den Daten *keine* Evidenz gibt. In diesem Sinne sind es plausible bzw. „gute“ Modelle.

MAXENT–Algorithmus

Es seien m Funktionen $\varphi_j(\boldsymbol{x})$, $j = 1, \ldots, m$ (diese werden auch als Indikatorfunktionen, Merkmalsfunktionen, “feature functions” bezeichnet) gegeben. Die *bekannten* Nebenbedingungen haben die Form

$$d_j = \sum_{i=1}^{n} p_i \varphi_j(\boldsymbol{x}_i) \;, \quad j = 1, \ldots, m \;, \tag{4.2.86}$$

d. h. sie sind *Mittelwerte* der Indikatorfunktionen, wie in (4.2.79). Es wird eine Partitionsfunktion

$$z(\vartheta_1, \ldots, \vartheta_m) = \sum_{i=1}^{n} \exp\left[- \sum_{j=1}^{m} \vartheta_j \varphi_j(\boldsymbol{x}_i) \right] \tag{4.2.87}$$

mit LAGRANGE-Multiplikatoren ϑ_j definiert. Die Wahrscheinlichkeiten p_i, die die Entropie maximieren, sind gegeben durch folgenden Satz:

Satz 4.9 *Die Wahrscheinlichkeiten p_i mit maximaler Entropie ergeben sich aus*

$$p_i = \frac{1}{z} \exp\left[- \sum_{j=1}^{m} \vartheta_j \varphi_j(\boldsymbol{x}_i) \right] \;, \quad i = 1, \ldots, n \;. \tag{4.2.88}$$

Beweis*: s. z. B. [Gibbs, 1905, Jaynes, 1982], bzw. (4.2.103) und (4.2.104).*

Damit ist die maximale Entropie gegeben durch

$$H_{\max} = \ln[z] + \sum_{j=1}^{m} \vartheta_j d_j \;, \tag{4.2.89}$$

und die LAGRANGE-Multiplikatoren ergeben sich aus

$$\frac{\partial \ln z}{\partial \vartheta_j} + d_j = 0 \;, \quad j = 1, \ldots, m \;. \tag{4.2.90}$$

MAXENT–Modellierung

Die generelle Vorgehensweise bei der Erstellung eines stochastischen Modells auf der Basis der Entropiemaximierung folgt den obigen Ausführungen. Es wird eine Menge

$$\omega = \{(x_1, y_1), \ldots, (x_\varrho, y_\varrho), \ldots, (x_N, y_N)\} \tag{4.2.91}$$

von Trainingsdaten gesammelt. Dabei sind die y_ϱ Werte einer Zufallsvariablen aus einer endlichen Menge, und die x_ϱ sind Werte einer (ggf. vektorwertigen) Zufallsvariablen, die Einfluss auf den beobachteten Wert y_ϱ haben. Gesucht ist ein stochastisches Modell, das eine Schätzung des Wertes von y erlaubt, wenn x gegeben ist, d. h. gesucht ist die bedingte Verteilung $p(y|x)$. Identifiziert man y mit Ω_κ und x mit $\boldsymbol{c}$, so liegt gerade der Fall (4.1.3), S. 306, vor. Identifiziert

man y mit der Übersetzung eines Wortes aus einer Sprache S_1 (z. B. des englischen Wortes "to") in eine Sprache S_2 (z. B. Deutsch), so sind mögliche Werte von y z. B. „nach" und „zu"; x ist dann der Kontext, in dem das Wort beobachtet wurde. Wenn die Menge der Werte von x und y endlich ist, lassen sich durch Abzählen die zwei Wertetabellen bestimmen

$$N_{x,y} = \sum_{\varrho=1}^{N} \delta(x_\varrho, x)\delta(y_\varrho, y) \ , \quad \text{(Anzahl der Werte von } (x,y) \text{ in } \omega) \ , \tag{4.2.92}$$

$$N_x = \sum_y N_{x,y} \ , \quad \text{(Anzahl der Werte von } (x) \text{ in } \omega) \ . \tag{4.2.93}$$

Daraus ergeben sich die empirische Verbunddichte $\widehat{p}(x,y)$ sowie die empirische Dichte $\widehat{p}(x)$

$$\widehat{p}(x,y) = \frac{N_{x,y}}{N} \ , \qquad \widehat{p}(x) = \sum_y \widehat{p}(x,y) \ . \tag{4.2.94}$$

Das Ziel besteht darin, eine Modelldichte $p(y|x)$ zu konstruieren, die die Stichprobe ω, repräsentiert durch die empirische Dichte $\widehat{p}(x,y)$, generiert *und* eine Menge von Nebenbedingungen erfüllt. Diese können die Form (4.2.79) haben bzw. irgendeine Statistik im Sinne von (4.2.64) der Stichprobe sein. Zu diesem Zweck werden binäre **Indikatorfunktionen**

$$\varphi_j(x_\varrho, y_\varrho) = \begin{cases} 1 & : \quad x = x_\varrho \wedge y = y_\varrho \\ 0 & : \quad \text{sonst} \end{cases} \tag{4.2.95}$$

eingeführt. Als Nebenbedingungen werden z. B. die Erwartungswerte dieser Indikatorfunktionen bezüglich der empirischen Dichte verwendet

$$d_j = \sum_{x,y} \widehat{p}(x,y)\varphi_j(x,y) \ , \quad \text{bzw. } N_j = \sum_{x,y} N_{x,y}\varphi_j(x,y) \ . \tag{4.2.96}$$

Der Erwartungswert der Indikatorfunktion bezüglich der Modelldichte ist

$$E\{\varphi_j\} = \sum_{x,y} \widehat{p}(x)p(y|x)\varphi_j(x,y) \ , \tag{4.2.97}$$

und es wird gefordert, dass

$$d_j = E\{\varphi_j\} \tag{4.2.98}$$

gilt. Es kommen also nur Modelldichten in Frage, die (4.2.98) genügen. Das Prinzip der Entropiemaximierung besagt nun, dass unter den in Frage kommenden Dichten die mit maximaler (bedingter) Entropie ausgewählt wird

$$p^*(y|x) = \underset{\{p\}}{\operatorname{argmax}}\, H(p) = \underset{\{p\}}{\operatorname{argmax}} \left\{ -\sum_{x,y} \widehat{p}(x)p(y|x)\ln p(y|x) \right\} \ . \tag{4.2.99}$$

Es lässt sich zeigen, dass diese Aufgabe der Optimierung mit Nebenbedingungen (s. Abschnitt 1.6.3) eine *eindeutige* Lösung für p^* unter den genannten Bedingungen hat. Die Nebenbedingungen sind zusammengefasst

$$0 \ \leq \ p(y|x) \ , \quad \forall (x,y) \ ,$$

$$1 = \sum_y p(y|x) \;, \quad \forall x \;, \tag{4.2.100}$$

$$d_j = \sum_{x,y} \widehat{p}(x,y)\varphi_j(x,y) = \sum_{x,y} \widehat{p}(x)p(y|x)\varphi_j(x,y) \;, \quad j = 1,\ldots,m \;. \tag{4.2.101}$$

Die LAGRANGE-Gleichung dafür ist

$$\begin{aligned} L(p,\vartheta_{0,x},\vartheta_j) &= -\sum_{x,y} \widehat{p}(x)p(y|x)\ln p(y|x) - \sum_x \vartheta_{0,x}\left(\sum_y p(y|x) - 1\right) \\ &\quad - \sum_{j=1}^m \vartheta_j \left(\sum_{x,y} \widehat{p}(x,y)\varphi_j(x,y) - \sum_{x,y} \widehat{p}(x)p(y|x)\varphi_j(x,y)\right) \end{aligned} \tag{4.2.102}$$

wobei die $\vartheta_{0,x}$ LAGRANGE-Multiplikatoren sind, die für die für *jeden* Wert von x zu erfüllenden Nebenbedingungen (4.2.100) eingeführt wurden; ϑ_j sind solche, die für jede der Merkmalsfunktionen (4.2.101) eingeführt wurden. Die partielle Ableitung nach $p(y|x)$ ergibt

$$\frac{\partial L}{\partial p(y|x)} = -\widehat{p}(x)\left(1 + \ln[p(y|x)]\right) + \sum_j \vartheta_j \widehat{p}(x)\varphi_j(x,y) - \vartheta_{0,x} \;, \tag{4.2.103}$$

$$\begin{aligned} p^*(y|x) &= \exp\left[-\frac{\vartheta_{0,x}}{\widehat{p}(x)} - 1\right] \exp\left[\sum_{j=1}^m \vartheta_j\varphi_j(x,y)\right] \\ &= z(x)\exp\left[\sum_{j=1}^m \vartheta_j\varphi_j(x,y)\right] \;. \end{aligned} \tag{4.2.104}$$

Es ergibt sich also auch hier wieder eine Exponentialfunktion wie in (4.2.88). Der normierende Faktor $z(x)$ ist

$$z(x) = \left(\sum_y \exp\left[\sum_j \vartheta_j\varphi_j(x,y)\right]\right)^{-1} \;. \tag{4.2.105}$$

Er ergibt sich, indem man zunächst (4.2.104) in (4.2.100) einsetzt und daraus $\vartheta_{0,x}$ bestimmt zu

$$\vartheta_{0,x} = -\widehat{p}(x) + \widehat{p}(x)\ln\left[\sum_y \exp\left[\sum_{j=1}^m \vartheta_j\varphi_j(x,y)\right]\right] \;; \tag{4.2.106}$$

dann wird der Wert für $\vartheta_{0,x}$ in $z(x)$ in (4.2.104) eingesetzt. So erhält man schließlich

$$\boxed{p^*(y|x) = \frac{\exp\left[\sum_{j=1}^m \vartheta_j\varphi_j(x,y)\right]}{\sum_y \exp\left[\sum_{j=1}^m \vartheta_j\varphi_j(x,y)\right]} = z(x)\exp\left[\sum_{j=1}^m \vartheta_j\varphi_j(x,y)\right] \;.} \tag{4.2.107}$$

GIS–Algorithmus

Die Abkürzung GIS steht für "Generalized Iterative Scaling" und bezeichnet einen iterativen Ansatz zur Bestimmung der noch fehlenden LAGRANGE-Multiplikatoren ϑ_j. Die iterative Lö-

berechne Merkmalszahlen N_i; wähle Startwerte für $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_j, \ldots, \vartheta_m)$
FOR jeden Interationsschritt:
  initialisiere $g(\boldsymbol{\vartheta}) = \mathbf{0}$, initialisiere $q_j(\boldsymbol{\vartheta}) = \mathbf{0}$, $j = 1, \ldots, m$
  FOR jedes Stichprobenelement x_ϱ, $\varrho = 1, \ldots, N$:
    berechne Normierungsfaktor $z(x_\varrho)$
    ersetze $g = g + \log p^*(y_\varrho|x_\varrho)$
    FOR jede Klasse:
      FOR jede Indikatorfunktion:
        $q_j = q_j + p^*(y|x_\varrho)$
  $\vartheta_j = \vartheta_j + \Delta\vartheta_j$

Bild 4.2.4: GIS–Algorithmus zur Bestimmung der LAGRANGE-Multiplikatoren ϑ_j in (4.2.107)

sung ist notwendig, da i. Allg. keine geschlossene möglich ist. Wir geben hier die Grundzüge des Algorithmus in Bild 4.2.4 an und verweisen für Einzelheiten auf die Literatur.

Es lässt sich zeigen, dass man von dem *dualen Problem* der Maximierung von

$$\begin{aligned} g(\boldsymbol{\vartheta}) &= \sum_{\varrho=1}^{N} \log\left[p^*(y_\varrho|x_\varrho)\right] \\ &= \sum_{x,y} N_{x,y} \log\left[p^*(y|x)\right] \end{aligned} \tag{4.2.108}$$

bezüglich $\boldsymbol{\vartheta}$ ausgehen kann. Die partiellen Ableitungen von g nach ϑ_j ergeben

$$\frac{\partial g}{\partial \vartheta_j} = N_j - q_j(\boldsymbol{\vartheta}) \;, \tag{4.2.109}$$

$$N_j = \sum_{x,y} N_{x,y} \varphi_j(x,y) \;, \tag{4.2.110}$$

$$\begin{aligned} q_j &= \sum_x N_x \sum_y p^*(y|x)\varphi_j(x,y) \\ &= \sum_{\varrho=1}^{N} \sum_y p^*(y|x_\varrho)\varphi_j(x_\varrho, y) \;. \end{aligned} \tag{4.2.111}$$

Die LAGRANGE-Multiplikatoren lassen sich nun iterativ aus

$$\vartheta_j^{\nu+1} = \vartheta_j^\nu + \Delta\vartheta_j \tag{4.2.112}$$

bestimmen, wobei ν den Iterationsschritt angibt. Die Korrekturterme $\Delta\vartheta_j$ erhält man als Lösung der Gleichung

$$N_j = \sum_{x,y} N_x p^*(y|x)\varphi_j(x,y) \exp\left[\Delta\vartheta_j \sum_i \varphi_i(x,y)\right] \;. \tag{4.2.113}$$

Wenn zudem die Zahl der sogenannten aktiven Indikatoren konstant ist, d. h.

$$\sum_j \varphi_j(x,y) = F_\varphi = \text{const} \tag{4.2.114}$$

ist, dann gibt es für $\Delta\vartheta_j$ die *geschlossene Lösung*

$$\Delta\vartheta_j = \frac{1}{F_\varphi} \log \frac{N_j}{q_j(\boldsymbol{\vartheta})} . \tag{4.2.115}$$

4.2.5 Klassifikation normalverteilter Merkmalsvektoren

Wie bereits in Abschnitt 4.2.1 erwähnt, bilden die n–dimensionalen Normalverteilungsdichten die wichtigste parametrische Familie. Daher wird in diesem Abschnitt etwas genauer der Fall klassenweise normalverteilter Merkmalsvektoren betrachtet; der resultierende Klassifikator wird auch als **Normalverteilungsklassifikator** bezeichnet. Im Allgemeinen ergeben sich die Prüfgrößen durch Einsetzen von (4.2.3) in (4.1.13). Für die (0,1)–Kostenfunktion erhält man eine weitere Vereinfachung. Als Prüfgrößen werden die in (4.1.33) auftretenden Terme

$$\begin{aligned} u_\lambda(\boldsymbol{c}) &= p_\lambda p(\boldsymbol{c}|\Omega_\lambda) \\ &= p_\lambda \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_\lambda|}} \exp\left[-\frac{(\boldsymbol{c}-\boldsymbol{\mu}_\lambda)^\mathsf{T}\boldsymbol{\Sigma}_\lambda^{-1}(\boldsymbol{c}-\boldsymbol{\mu}_\lambda)}{2}\right] \end{aligned} \tag{4.2.116}$$

verwendet. Die Lage des in (4.1.33) zu ermittelnden Maximum bezüglich λ ändert sich nicht, wenn eine monoton wachsende Funktion von $u_\lambda(\boldsymbol{c})$ genommen wird. In diesem Falle ist

$$u'_\lambda(\boldsymbol{c}) = 2\ln\left[u_\lambda(\boldsymbol{c})\right] \tag{4.2.117}$$

zweckmäßig. Damit erhält man

$$\begin{aligned} u'_\lambda(\boldsymbol{c}) &= -(\boldsymbol{c}-\boldsymbol{\mu}_\lambda)^\mathsf{T}\boldsymbol{\Sigma}_\lambda^{-1}(\boldsymbol{c}-\boldsymbol{\mu}_\lambda) + 2\ln\left(\frac{p_\lambda}{\sqrt{|2\pi\boldsymbol{\Sigma}_\lambda|}}\right) \\ &= -\boldsymbol{c}^\mathsf{T}\boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{c} + 2\boldsymbol{c}^\mathsf{T}\boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{\mu}_\lambda + \gamma_\lambda , \\ \gamma_\lambda &= -\boldsymbol{\mu}_\lambda^\mathsf{T}\boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{\mu}_\lambda + 2\ln\left(\frac{p_\lambda}{\sqrt{|2\pi\boldsymbol{\Sigma}_\lambda|}}\right) \end{aligned} \tag{4.2.118}$$

Definiert man einen Vektor mit $(1+n+n(n+1)/2)$ Komponenten

$$\widetilde{\boldsymbol{c}}^\mathsf{T} = (1, c_1, c_2, \ldots, c_n, c_1c_1, c_2c_1, c_2c_2, c_3c_1, \ldots, c_nc_n) , \tag{4.2.119}$$

einen Vektor mit n Komponenten

$$\boldsymbol{a}_\lambda = 2\boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{\mu}_\lambda , \tag{4.2.120}$$

bezeichnet die Elemente der symmetrischen Matrix $\boldsymbol{\Sigma}_\lambda^{-1}$ mit $\sigma_{\lambda ij}$, $i,j = 1,\ldots,n$, und definiert einen Vektor

$$\widetilde{\boldsymbol{a}}_\lambda^\mathsf{T} = (\gamma_\lambda, a_{\lambda 1}, \ldots, a_{\lambda n}, -\sigma_{\lambda 11}, -2\sigma_{\lambda 21}, -\sigma_{\lambda 22}, -2\sigma_{\lambda 31}, \ldots, -\sigma_{\lambda nn}) , \tag{4.2.121}$$

so gilt

$$u'_\lambda(\boldsymbol{c}) = \widetilde{\boldsymbol{a}}_\lambda^\mathsf{T}\widetilde{\boldsymbol{c}} , \quad \lambda = 1,\ldots,k . \tag{4.2.122}$$

Die Anwendung von (4.1.33) auf normalverteilte Merkmalsvektoren erfordert also die Berechnung von k Skalarprodukten. Der resultierende Klassifikator, dessen Struktur in Bild 4.2.5 angegeben ist, ist *quadratisch* bezüglich der Komponenten c_ν des Merkmalsvektors. Der Vektor

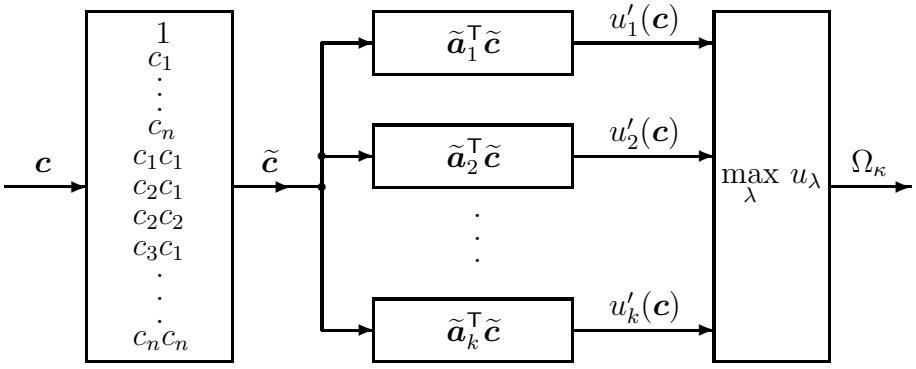


Bild 4.2.5: Die Struktur des Klassifikators, der für normalverteilte Merkmale die Fehlerwahrscheinlichkeit minimiert, gemäß (4.1.33), S. 314, (4.2.122)

$\widetilde{\boldsymbol{c}}$ ist für alle k Prüfgrößen gleich, die Vektoren $\widetilde{\boldsymbol{a}}_\lambda$ werden in der Lern- oder Trainingsphase des Klassifikators berechnet, als Parameter gespeichert und dann nicht mehr oder nur selten verändert.

Gemäß (4.2.116) liegen Punkte mit gleichen Werten von u_λ auf Hyperellipsoiden des $\mathbb{R}_{\boldsymbol{c}}$. Wenn man nur zwei Komponenten c_ν, c_μ des Merkmalsvektors betrachtet, ergeben sich Ellipsen, die man zur Veranschaulichung der Verhältnisse graphisch darstellen kann. Die **Trennfläche** zwischen zwei Klassen Ω_κ und Ω_λ ergibt sich aus der Gleichung

$$u_\lambda(\boldsymbol{c}) = u_\kappa(\boldsymbol{c}) \ . \tag{4.2.123}$$

Ein Beispiel wurde bereits in Bild 3.8.7 gezeigt.

Sind die bedingten Kovarianzmatrizen Diagonalmatrizen

$$\boldsymbol{\Sigma}_\lambda = \text{diag}(\sigma_{\lambda 1}, \sigma_{\lambda 2}, ..., \sigma_{\lambda n}) \ , \tag{4.2.124}$$

so erhält man für die Prüfgrößen aus (4.2.118)

$$u'_\lambda(\boldsymbol{c}) = -\sum_{\nu=1}^{n} \frac{(c_\nu - \mu_{\lambda\nu})^2}{\sigma_{\lambda\nu}} + 2\ln[p_\lambda] - \sum_{\nu=1}^{n} \ln[2\pi\sigma_{\lambda\nu}] \ . \tag{4.2.125}$$

Im Wesentlichen handelt es sich also um einen bewichteten Abstand des Merkmalsvektors vom Klassenzentrum $\boldsymbol{\mu}_\lambda$. Da (4.2.125) numerisch wesentlich einfacher zu berechnen ist als (4.2.122), wird diese Form oft auch als suboptimaler Klassifikator verwendet. Vielfach wird sogar nur der EUKLID-Abstand

$$u'_\lambda(\boldsymbol{c}) = -\sum_{\nu=1}^{n} (c_\nu - \mu_{\lambda\nu})^2 \tag{4.2.126}$$

verwendet. Da man (4.2.126) auch in der Form

$$u'_\lambda(\boldsymbol{c}) = -\sum_{\nu=1}^{n} c_\nu^2 + \sum_{\nu=1}^{n} (2c_\nu\mu_{\lambda\nu} - \mu_{\lambda\nu}^2) \tag{4.2.127}$$

schreiben kann, ist die Lage des Maximums der Prüfgrößen von $\sum c_\nu^2$ unabhängig, sodass sich eine weitere Vereinfachung der Berechnung auf Prüfgrößen, die *linear* bezüglich c_ν sind, ergibt. Wenn man bereits mit den einfacheren Formen (4.2.127) oder (4.2.125) zufriedenstellende

Klassifikationsergebnisse erhält, besteht natürlich kein Grund, die aufwendiger zu berechnenden zu verwenden. Ein verbessertes Abstandsmaß, den MAHALANOBIS-Abstand, erhält man, wenn man in (4.2.118) die Größe $2\ln(\cdot)$ vernachlässigt. Der zugehörige Klassifikator wurde bereits in Abschnitt 3.8.4 als (modifizierter) **Minimumabstandsklassifikator** (MMA) eingeführt.

Wenn alle bedingten Kovarianzmatrizen gleich sind, also

$$\boldsymbol{\Sigma}_\lambda = \boldsymbol{\Sigma}\ , \tag{4.2.128}$$

erhält man in (4.2.118) Prüfgrößen, bei denen die Lage des Maximums unabhängig vom Term $\boldsymbol{c}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{c}$ ist. Es genügen Prüfgrößen

$$u''_\lambda(\boldsymbol{c}) = 2\boldsymbol{c}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_\lambda + \gamma_\lambda, \tag{4.2.129}$$

die *linear* in den Komponenten c_ν des Merkmalsvektors sind.

Die Anwendung der Kostenfunktion (4.1.19), S. 311, erfordert zur Auswertung des Rückweisungskriteriums (4.1.25) die Berechnung des Terms $\alpha(\boldsymbol{c})$ gemäß (4.1.23). Die Logarithmierung wie in (4.2.117) bringt hier keine Vorteile, vielmehr müssen tatsächlich die Exponentialfunktionen in (4.2.116) berechnet werden. Erfahrungsgemäß können dabei Größenordnungen von Zahlen auftreten, die außerhalb des üblichen Gleitkommabereichs eines Rechners liegen. Ein problemlos zu berechnendes heuristisches Rückweisungskriterium ist das Folgende. Man berechne die k Prüfgrößen $u'_\lambda(\boldsymbol{c})$ gemäß (4.2.118) oder (4.2.122) und ermittle die größte und zweitgrößte

$$u_{\kappa 1} = \max_\lambda u_\lambda \quad \text{und} \quad u_{\kappa 2} = \max_{\lambda\neq\kappa 1} u_\lambda\ . \tag{4.2.130}$$

Das Muster wird zurückgewiesen, wenn

$$\frac{(u_{\kappa 1} - u_{\kappa 2})}{u_{\kappa 1}} < \theta \tag{4.2.131}$$

ist, und sonst der Klasse $\Omega_{\kappa 1}$ zugeordnet.

Die zentrale Annahme dieses Abschnittes, nämlich klassenweise normalverteilte Merkmalsvektoren, wird i. Allg. nur eine Approximation der realen Verhältnisse sein. Die Berechtigung dieser Annahme lässt sich auf zwei Arten überprüfen:

1. Man realisiert einen Klassifikator auf der Basis dieser Annahme und ermittelt experimentell, d. h. durch Klassifikation einer genügend großen Stichprobe, seine Leistungsfähigkeit, z. B. mit (4.1.40) oder (3.9.9). Wenn die Leistung ausreicht, ist auch die Annahme normalverteilter Merkmale ausreichend genau.
2. Man testet alle eindimensionalen marginalen Dichten von $p(\boldsymbol{c}|\Omega_\kappa)$ auf Normalverteilung. Das kann z. B. mit einem KOLMOGOROW–SMIRNOW-Test erfolgen.

Wenn eine oder einige der Komponenten von $\boldsymbol{c}$ nicht normalverteilt sind, ist die Annahme normalverteilter Merkmale sicher unzutreffend. Wenn alle Komponenten von $\boldsymbol{c}$ normalverteilt sind, also alle eindimensionalen marginalen Dichten von $p(\boldsymbol{c})|\Omega_\kappa)$, dann ist das zwar ein Indiz für die Richtigkeit der Annahme, aber kein Beweis, da sich Gegenbeispiele konstruieren lassen.

Wenn die Untersuchungen nach Punkt 1 und/oder Punkt 2 negativ ausfielen, hat man drei Möglichkeiten:

1. Man versucht, eine andere parametrische Familie zu finden (dieses hat bisher geringe praktische Bedeutung erlangt) oder die Anwendung von (4.2.13) oder (4.2.15).

2. Man versucht, andere Merkmale zu finden, für die die Annahme der Normalverteilung besser zutrifft.
3. Man untersucht, ob ein anderer Klassifikator, wie die in den folgenden Abschnitten beschriebenen, bessere Ergebnisse liefert.

Die hier aufgezeigten Möglichkeiten erfordern einen hohen experimentellen Aufwand. Das mag für den elegante geschlossene Lösungen suchenden Theoretiker unbefriedigend und für den schnelle Erfolge erwartenden Anwender unrealistisch erscheinen. Es gibt aber nicht einige wenige mit einer kleinen Stichprobe einfach zu schätzende Parameter, mit denen sich eine Aussage machen ließe, welche Merkmale und welcher Klassifikator der beste ist; bei der Komplexität des Problems ist das auch nicht zu erwarten. Sorgfältige und entsprechend aufwendige experimentelle Untersuchungen sind daher unerlässlich. Natürlich wird man dabei bereits bekannte experimentelle Ergebnisse berücksichtigen und nicht erneut erarbeiten.

Die Realisierung eines Klassifikators gemäß (4.1.33) und (4.2.116) unter der Annahme normalverteilter Merkmale ist relativ einfach, vorausgesetzt es steht eine klassifizierte Stichprobe mit $N_\kappa > n$ Mustern je Klasse zur Verfügung. Auch wenn die Merkmale nur näherungsweise normalverteilt sind, gibt es erfahrungsgemäß verschiedene praktisch interessante Aufgaben, bei denen dieser Klassifikator gute Ergebnisse liefert.

4.2.6 Nichtparametrische Schätzung von Verteilungsdichten

In Abschnitt 4.2 wurde mit (4.1.1) vorausgesetzt, dass bestimmte statistische Vorkenntnisse in Form einer parametrischen Familie von Verteilungsdichten gegeben sind. Mit Hilfe nichtparametrischer statistischer Verfahren ist es möglich, Verteilungsdichten zu schätzen, ohne solche Vorkenntnisse zu verlangen.

> Als *nichtparametrisches statistisches Verfahren* bezeichnet man eines, das für viele, wenn möglich für *alle*, Familien von Verteilungsdichten gültig ist.

Der Preis dafür ist meistens, wie sich noch zeigen wird, dass die gesamte Stichprobe ω gespeichert werden muss. Bei den Klassifikatoren von Abschnitt 4.2.5 und Abschnitt 4.4 genügt die Speicherung der Parametervektoren, deren Größe insbesondere *unabhängig* vom Umfang der Stichprobe ist. Die praktische Bedeutung dieser nichtparametrischen Schätzungen ist daher begrenzt, sodass nur kurz auf sie eingegangen wird. Als Beispiele für **nichtparametrische Schätzungen** werden die direkte Schätzung und die PARZEN-Schätzung behandelt. Die direkte Schätzung beruht darauf, dass man einen Schätzwert $\widehat{p}(\boldsymbol{c}|\Omega_\kappa)$ für die Verteilungsdichte im Punkt $\boldsymbol{c}$ erhält aus

$$\widehat{p}(\boldsymbol{c}|\Omega_\kappa) = \frac{P_\kappa}{V} \,, \tag{4.2.132}$$

wobei V ein bestimmtes Volumen des $\mathbb{R}^n$ ist und $\boldsymbol{c}$ enthält, und P_κ ist die Wahrscheinlichkeit, dass Merkmalsvektoren aus Ω_κ im Volumen V liegen. Ist eine Stichprobe von Mustern aus Ω_κ vom Umfang N_κ gegeben und liegen m_κ Muster in V, so ist ein Schätzwert für P_κ

$$\widehat{P}_\kappa = \frac{m_\kappa}{N_\kappa} \,. \tag{4.2.133}$$

Als Schätzwert der bedingten Dichte wird nun

$$\widehat{p}(\boldsymbol{c}|\Omega_\kappa) = \frac{\widehat{P}_\kappa}{V} = \frac{m_\kappa}{N_\kappa V} \tag{4.2.134}$$

verwendet. Bei der Schätzung mit Histogrammen (s. Abschnitt 2.2.2 und Abschnitt 4.9.2) ist $V = \text{const}$, d. h. man teilt den interessierenden Bereich des Merkmalsraumes in (meistens) gleichgroße n-dimensionale Intervalle und bestimmt dann die Zahl m_κ der Muster je Intervall. Wenn man die Intervalle fest vorgibt, brauchen nur die Intervallgrenzen und die dafür ermittelten Werte $\widehat{P}_\kappa/V$ gespeichert zu werden; die Speicherung der Stichprobe ist nicht erforderlich. Die Wahl der Intervalle ist dabei ein Problem, das nicht befriedigend gelöst ist. Es kann insbesondere passieren, dass sich bei Intervallen, die im Verhältnis zum Stichprobenumfang zu klein sind, öfters der Schätzwert Null ergibt. Das lässt sich vermeiden, wenn man $m_\kappa = \text{const}$ setzt und V variabel lässt. Ein Schätzwert der Dichte im Punkte $\boldsymbol{c}$ wird dadurch bestimmt, dass man die $(m_\kappa + 1)$ nächsten Nachbarn von $\boldsymbol{c}$ sucht. Der am weitesten entfernte dieser Nachbarn habe von $\boldsymbol{c}$ den Abstand r. Man kann sich nun vorstellen, dass in einer Hyperkugel mit Mittelpunkt $\boldsymbol{c}$ und Radius r die $m_\kappa + 1$ Stichprobenelemente liegen, von denen das am weitesten entfernte auf der Kugeloberfläche liegt. Ein Schätzwert für die Dichte im Punkt $\boldsymbol{c}$ ist dann durch (4.2.134) gegeben, wobei für V das Volumen der Hyperkugel

$$V = \frac{2r^n \pi^{n/2}}{n\Gamma\left(\frac{n}{2}\right)} \tag{4.2.135}$$

eingesetzt wird. Unter den Voraussetzungen

$$\lim_{N_\kappa \to \infty} m_\kappa(N_\kappa) = \infty \,, \qquad \lim_{N_\kappa \to \infty} \frac{m_\kappa(N_\kappa)}{N_\kappa} = 0 \tag{4.2.136}$$

konvergiert der Schätzwert. Gemäß (4.2.136) muss m_κ vom Stichprobenumfang abhängen, eine geeignete Wahl ist

$$m_\kappa = \sqrt{N_\kappa} \,. \tag{4.2.137}$$

Bei dieser Schätzung ist eine vorherige feste Aufteilung des Merkmalsraumes nicht möglich, es muss also die gesamte Stichprobe gespeichert werden.

Eine andere Möglichkeit zur Schätzung einer Verteilungsdichte beruht darauf, dass man für jede Stichprobe

$$\omega_\kappa = \{{}^1\boldsymbol{c}_\kappa,\, {}^2\boldsymbol{c}_\kappa, \ldots,\, {}^{N_\kappa}\boldsymbol{c}_\kappa\} \tag{4.2.138}$$

unmittelbar eine **empirische Verteilungsdichte**

$$\widehat{p}_E(\boldsymbol{c}|\Omega_\kappa) = \frac{1}{N_\kappa} \sum_{j=1}^{N_\kappa} \delta(\boldsymbol{c} - {}^j\boldsymbol{c}_\kappa) \tag{4.2.139}$$

angeben kann. Praktisch ist diese Form wenig nützlich, da für (fast) alle neu beobachteten Muster $\widehat{p}_E = 0$ wird; es liegt aber die Vermutung nahe, dass man besser geeignete Schätzwerte erhält, wenn man die δ–Funktion (s. (2.1.9), S. 64) durch Fensterfunktionen (bzw. Potential– oder Kernfunktionen) ersetzt, die auch in einer gewissen Umgebung von ${}^j\boldsymbol{c}_\kappa$ von Null verschiedene Werte annehmen. Die resultierenden Schätzverfahren werden als PARZEN-Schätzung oder als kernbasierte Schätzung bezeichnet. Es sei

$$g_0(\boldsymbol{x}) = g_0\left(\frac{\boldsymbol{c} - {}^j\boldsymbol{c}}{h_N}\right) = g_0\left((\boldsymbol{c} - {}^j\boldsymbol{c})\,|\,h_N\right) \tag{4.2.140}$$

eine Fensterfunktion, deren Breite durch den Parameter h_N bestimmt wird und den Bedingungen

$$\begin{aligned} 0 &\leq g_0(\boldsymbol{x}) \,, \quad \int_{-\infty}^{\infty} g_0(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 1 \,, \\ 0 &= \lim_{|\boldsymbol{x}|\to\infty} g_0(\boldsymbol{x}) \prod_{\nu=1}^{n} x_\nu \,, \quad \sup g_0(\boldsymbol{x}) < \infty \,, \\ 0 &= \lim_{N\to\infty} h_N^n \,, \quad \lim_{N\to\infty} N h_N^n = \infty \end{aligned} \tag{4.2.141}$$

genügt.

Satz 4.10 *Die* PARZEN-**Schätzung**

$$\widehat{p}(\boldsymbol{c}|\Omega_\kappa) = \frac{1}{N_\kappa} \sum_{j=1}^{N_\kappa} g_0\left((\boldsymbol{c} - {}^j\boldsymbol{c}_{\boldsymbol{\kappa}})|h_N\right) \tag{4.2.142}$$

konvergiert im quadratischen Mittel gegen die Dichte $p(\boldsymbol{c}|\Omega_\kappa)$, wenn diese an der Stelle $\boldsymbol{c}$ stetig ist.
Beweis*: s. z. B. [Parzen, 1962, Murthy, 1965].*

Mögliche Fensterfunktionen sind z. B. das Rechteckfenster und die GAUSS-Funktion. Bei letzterer wird oft, wie auch in (4.8.37), S. 426, die Kovarianzmatrix vereinfacht zu $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$, womit sich $h_N = \sigma^2$ ergibt; zur Verwendung einer allgemeinen Matrix $\boldsymbol{\Sigma}$ bei der Schätzung wird auf die Literatur verwiesen. Auch bei dieser Schätzung muss die gesamte Stichprobe gespeichert werden. Eine Reduktion oder Verdünnung, z. B. durch Ersetzung der Stichprobe durch eine genügend große Anzahl von Kodebuchvektoren einer Vektorquantisierung oder durch ein anderes Verfahren zur Analyse von Häufungsgebieten (s. Abschnitt 4.8.4, „Clusteranalyse“), kann dieses Problem mildern. Es ist weiterhin möglich, auf dieser Basis auch die Moden (relativen Extrema) einer Verteilungsdichte zu schätzen. Durch Einsatz von Optimierungsverfahren kann die Stichprobe reduziert und damit die Schätzung vereinfacht werden. Auch dafür wird auf die Literatur verwiesen.

4.2.7 Nächster Nachbar Klassifikator

Der nächste Nachbar (NN) Klassifikator beruht darauf, ein Muster der Klasse zuzuordnen, zu der auch der nächste Nachbar im Merkmalsraum bzw. die Mehrzahl seiner m nächsten Nachbarn gehören. Dieser zunächst rein intuitiv naheliegende Ansatz lässt sich auch als Schätzung der a posteriori Wahrscheinlichkeiten $p(\Omega_\kappa|\boldsymbol{c})$ auffassen und damit auf Satz 4.3 zurückführen. Ersetzt man in (4.1.34) die bedingte Dichte durch den Schätzwert (4.2.134), die a priori Wahrscheinlichkeit durch den Schätzwert

$$\widehat{p}_\kappa = \frac{N_\kappa}{N} \tag{4.2.143}$$

und die Dichte $p(\boldsymbol{c})$ analog zu (4.2.134) durch den Schätzwert

$$\widehat{p}(\boldsymbol{c}) = \frac{m}{NV} \,, \tag{4.2.144}$$

wobei N der Umfang der Stichprobe ω und m die Zahl der Muster im Volumen V ist, so ist ein Schätzwert der a posteriori Wahrscheinlichkeit

$$\widehat{p}(\Omega_\kappa|\boldsymbol{c}) = \frac{\widehat{p}_\kappa\,\widehat{p}(\boldsymbol{c}|\Omega_\kappa)}{\widehat{p}(\boldsymbol{c})} = \frac{m_\kappa}{m}\ . \tag{4.2.145}$$

Um zuverlässige Schätzungen zu erhalten, müssen m_κ und m genügend groß sein. Es lässt sich aber zeigen, dass schon der nächste Nachbar, also *nur ein* Stichprobenelement, eine recht zuverlässige Klassifikation erlaubt.

Zur Durchführung der NN Klassifikation wird eine *beliebige* **Metrik** $d(\boldsymbol{c},\,{}^j\boldsymbol{c})$ gewählt, mit der der Abstand eines neuen Musters $\boldsymbol{c}$ von einem Stichprobenelement ${}^j\boldsymbol{c}$ gemessen wird. Eine Klasse von Metriken sind beispielsweise

$$d^{(r)}(\boldsymbol{c},\,{}^j\boldsymbol{c}) = \left(\sum_{\nu=1}^{n} |c_\nu - {}^jc_\nu|^r\right)^{\frac{1}{r}}\ , \quad r = 1, 2, \ldots\ . \tag{4.2.146}$$

Bekannte Spezialfälle sind für $r = 1$ die *Cityblock Metrik*, für $r = 2$ der EUKLID-*Abstand* und für $r = \infty$ die *Maximumnorm*. Die NN Klassifikation arbeitet nach der Vorschrift

$$\text{wenn } d(\boldsymbol{c},\,{}^\varrho\boldsymbol{c}) = \min_j d(\boldsymbol{c},\,{}^j\boldsymbol{c}) \text{ und } {}^\varrho\boldsymbol{c} \in \omega_\kappa\ \text{, dann entscheide } \boldsymbol{c} \in \Omega_\kappa\ . \tag{4.2.147}$$

Ein Beispiel ist in Bild 4.2.6 gezeigt. Die Fehlerwahrscheinlichkeit p_f dieser NN–Regel kann relativ zur Fehlerwahrscheinlichkeit p_B des optimalen BAYES-Klassifikators (4.1.33) abgeschätzt werden, wobei diese Abschätzung *unabhängig* von der bedingten Verteilungsdichte der Merkmalsvektoren ist.

Satz 4.11 *Unter sehr allgemeinen Voraussetzungen gilt für jede Metrik und nahezu beliebige bedingte Verteilungsdichten für $N \to \infty$ und k Klassen die Abschätzung*

$$p_\mathrm{B} \le p_\mathrm{f} \le p_\mathrm{B}\left(2 - p_\mathrm{B}\frac{k}{k-1}\right)\ . \tag{4.2.148}$$

Die obigen Grenzen sind so eng wie möglich.
Beweis*: s. z. B. [Duda und Hart, 1972b, Cover und Hart, 1967]*

Da obiger Satz für praktisch alle Familien von Verteilungsdichten gilt, enthält er eine im obigen Sinne echt nichtparametrische Aussage. Ist die Fehlerwahrscheinlichkeit klein, also $p_\mathrm{B} \ll 1$, so geht (4.2.148) *näherungsweise* über in

$$\boxed{p_\mathrm{B} \le p_\mathrm{f} \le 2p_\mathrm{B}\ .} \tag{4.2.149}$$

Wenn man statt des einen nächsten Nachbarn die gesamte Stichprobe vom Umfang $N \to \infty$ zur Klassifikation heranzieht, kann man also die Fehlerwahrscheinlichkeit bestenfalls noch halbieren. In diesem Sinne kann man sagen, dass die halbe Information im nächsten Nachbarn steckt. Es ist wichtig, dass (4.2.148) und daher auch (4.2.149) nur für *sehr großen* Stichprobenumfang gelten. Praktisch kann man die NN–Regel (4.2.147) natürlich nur für endliches N auswerten. Es ist zu vermuten, dass auch für kleine Stichproben die Fehlerrate der NN–Regel einen Anhaltspunkt für die mit irgendeinem Klassifikator erreichbare gibt. Während mit den parametrischen Klassifikatoren, z. B. dem Normalverteilungsklassifikator von Abschnitt 4.2 oder

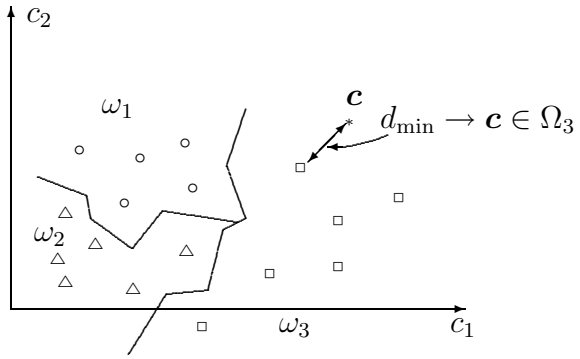


Bild 4.2.6: Zerlegung des Merkmalsraumes in Klassenbereiche durch die Nächster–Nachbar–Regel

dem Polynomklassifikator von Abschnitt 4.4, nur relativ einfache Trennflächen realisiert werden, ergeben sich bei der NN–Regel i. Allg. komplizierte nichtlineare Flächen, wie Bild 4.2.6 verdeutlicht. Verwendet man in (4.2.147) den EUKLID-Abstand, so gilt

$$d(\boldsymbol{c},\, {}^{j}\boldsymbol{c}) = \sqrt{(\boldsymbol{c} - {}^{j}\boldsymbol{c})^{\mathsf{T}}\,(\boldsymbol{c} - {}^{j}\boldsymbol{c})}\;. \tag{4.2.150}$$

Die Lage des Minimums von d bezüglich j ändert sich nicht, wenn in (4.2.147) statt d eine monoton zunehmende Funktion verwendet wird. Man kann also auch

$$d^2(\boldsymbol{c},\, {}^{j}\boldsymbol{c}) = \boldsymbol{c}^{\mathsf{T}}\,\boldsymbol{c} - 2\boldsymbol{c}^{\mathsf{T}}\,{}^{j}\boldsymbol{c} + {}^{j}\boldsymbol{c}^{\mathsf{T}}\,{}^{j}\boldsymbol{c} \tag{4.2.151}$$

berechnen. Definiert man Vektoren

$$\begin{aligned} \widetilde{\boldsymbol{c}}^{\mathsf{T}} &= (-0.5, c_1, c_2, \ldots, c_n) \\ {}^{j}\widetilde{\boldsymbol{c}}^{\mathsf{T}} &= ({}^{j}\boldsymbol{c}^{\mathsf{T}}\,{}^{j}\boldsymbol{c},\, {}^{j}c_1,\, {}^{j}c_2, \ldots,\, {}^{j}c_n)\;, \quad {}^{j}\boldsymbol{c} \in \omega\;, \end{aligned} \tag{4.2.152}$$

so kann man auch das Maximum von

$$u_j = {}^{j}\widetilde{\boldsymbol{c}}^{\mathsf{T}}\,\widetilde{\boldsymbol{c}}\;, \quad j = 1, \ldots, N \tag{4.2.153}$$

bezüglich j ermitteln und das neue Muster der Klasse zuordnen, die das Muster aus ω mit maximalem Wert von u_j hat.

Eine naheliegende Verbesserung der NN–Regel besteht darin, statt nur des nächsten Nachbarn die m nächsten Nachbarn zu bestimmen. Diese mNN–Regel arbeitet nach der Vorschrift

bestimme die m nächsten Nachbarn eines neuen Musters $\boldsymbol{c}$;
ordne $\boldsymbol{c}$ der Klasse zu, der die meisten der m Nachbarn angehören. (4.2.154)

Es lässt sich zeigen, dass für sehr große m und N die Fehlerwahrscheinlichkeit der mNN–Regel gegen die des BAYES-Klassifikators strebt. Ein guter Wert von m kann durch Verwendung einer von Trainings– und Teststichprobe disjunkten *Validierungsstichprobe* bestimmt werden. Praktisch werden oft Werte $m = 3$ bis $m = 13$ verwendet. Mit der mNN–Regel lassen sich also bessere Ergebnisse erzielen als mit der NN–Regel. Dieses ist wegen (4.2.145) plausibel, da der Schätzwert der a posteriori Wahrscheinlichkeit umso zuverlässiger wird, je größer m wird. Auch dieses Ergebnis gilt i. Allg. nur im Grenzfalle $N \rightarrow \infty$.

Initialisiere zwei Speicherbereiche SPEICHER und REST zur Aufnahme von Mustern; beide sind anfänglich leer.		
Bringe das erste Muster aus ω nach SPEICHER.		
FOR $i = 1$ TO N (N = Stichprobenumfang)		
	Klassifiziere mit der NN-Regel das i-te Muster aus ω nur unter Verwendung von Mustern aus SPEICHER.	
	IF	i-tes Muster richtig klassifiziert
	THEN	Bringe das Muster nach REST.
	ELSE	Bringe das Muster nach SPEICHER.
	FOR $i = 1$ TO N_R (N_R = Anzahl der Muster in REST)	
		Klassifiziere das i-te Muster aus REST nur unter Verwendung von Mustern aus SPEICHER.
		IF i-tes Muster nicht richtig klassifiziert
		THEN Entferne das Muster aus REST und bringe es nach SPEICHER.
UNTIL [Rest ist leer] ODER [es wurde kein Muster von REST nach SPEICHER gebracht]		

Bild 4.2.7: Berechnung einer „verdichteten“ Stichprobe für die NN–Regel

Die Einführung von *Rückweisungen* ist bei der mNN–Regel ebenfalls möglich. Dafür gibt es die zwei Vorschriften

bestimme die m nächsten Nachbarn eines neuen Musters $\boldsymbol{c}$;
wenn alle m Nachbarn aus Ω_κ sind,
dann klassifiziere $\boldsymbol{c}$ nach Ω_κ, sonst weise $\boldsymbol{c}$ zurück, (4.2.155)

bestimme die m nächsten Nachbarn eines neuen Musters $\boldsymbol{c}$;
wenn mindestens m' der m Nachbarn aus Ω_κ sind,
dann klassifiziere $\boldsymbol{c}$ nach Ω_κ, sonst weise $\boldsymbol{c}$ zurück. (4.2.156)

Für das Zweiklassenproblem lassen sich auch hier Abschätzungen der Fehlerwahrscheinlichkeiten angeben. Bei ungeraden Werten von m und $k = 2$ Klassen ist es nicht sinnvoll, $m' \leq (m+1)/2$ zu wählen, da es dann keine Rückweisungen geben kann.

Bei Anwendung der Regeln (4.2.147), (4.2.154) – (4.2.156) muss man die gesamte Stichprobe speichern und durchsuchen. Um den damit verbundenen Aufwand zu reduzieren, wurden verschiedene Vorschläge zur Verdichtung der Stichprobe gemacht, d. h. zur Eliminierung „unwichtiger“ Muster. Zum einen wird auf die Anmerkungen bei der PARZEN-Schätzung verwiesen, d. h. die Verwendung einer Vektorquantisierung oder eines anderen Ansatzes zur Analyse von Häufungsgebieten; zum anderen wurden spezielle Verfahren entwickelt, von denen einige kurz erwähnt werden. Jedes Muster aus einer klassifizierten Stichprobe ω wird mit der NN–Regel richtig klassifiziert. Wenn man nun Muster aus ω entfernen kann und immer noch alle Muster mit der verdichteten Stichprobe richtig klassifiziert werden, so ist diese bezüglich der Klassifikation äquivalent zu ω. Einen darauf basierenden Algorithmus zeigt Bild 4.2.7.

Am Ende des Algorithmus in Bild 4.2.7 enthält SPEICHER eine verdichtete Stichprobe, mit der ebenfalls alle Muster aus ω richtig klassifiziert werden. Ein Nachteil dieses Algorithmus ist, dass ein einmal in SPEICHER befindliches Muster nie mehr daraus entfernt wird. Im Allgemeinen wird daher SPEICHER keine minimale Stichprobe enthalten, d. h. es ist möglich,

Erzeuge aus ω eine Anfangsstichprobe $\omega_0 =$ SPEICHER; ω_0 enthalte N_S Muster.		
FOR $i = 1$ TO N_S		
	Klassifiziere alle Muster aus ω nur unter Verwendung von Mustern aus $\omega_0 - \{{}^i\boldsymbol{c}\}$	
	IF	alle Muster aus ω richtig klassifiziert
	THEN	Setze $\omega_0 = \omega_0 - \{{}^i\boldsymbol{c}\}$

Bild 4.2.8: Weitere Reduktion einer verdichteten Stichprobe

FOR $i = 1$ TO N_1			
	FOR $j = 1$ TO N_2		
		Berechne $\boldsymbol{m}_{ij} = ({}^i\boldsymbol{c}_1 + {}^j\boldsymbol{c}_2)/2$; ${}^i\boldsymbol{c}_1 \in \omega_1$, ${}^j\boldsymbol{c}_2 \in \omega_2$.	
		IF	[für alle ${}^\varrho\boldsymbol{c}_1 \in \omega_1$ gilt $\mathrm{d}(\boldsymbol{m}_{ij}\,,\,{}^\varrho\boldsymbol{c}_1) \geq \mathrm{d}(\boldsymbol{m}_{ij}\,,\,{}^i\boldsymbol{c}_1)$] AND [für alle ${}^\varrho\boldsymbol{c}_2 \in \omega_2$ gilt $\mathrm{d}(\boldsymbol{m}_{ij}\,,\,{}^\varrho\boldsymbol{c}_2) \geq \mathrm{d}(\boldsymbol{m}_{ij}\,,\,{}^j\boldsymbol{c}_2)$]
		THEN	Übernimm das Paar $\{{}^i\boldsymbol{c}_1, {}^j\boldsymbol{c}_2\}$ nach K–GRENZE

Bild 4.2.9: Bestimmung von Mustern auf der Klassengrenze, gespeichert in K–GRENZE

dass man auch mit weniger als den in SPEICHER befindlichen Mustern die gesamte Stichprobe ω richtig klassifizieren kann. Daher wird in Bild 4.2.8 ein erweiterter Algorithmus angegeben. Am Ende ist die reduzierte Untermenge in ω_0 enthalten. Alle Muster aus ω werden mit ω_0 richtig klassifiziert, aber am Ende enthält ω_0 i. Allg. weniger Muster als das anfängliche $\omega_0 =$ SPEICHER.

Auch zu dem zweiten Algorithmus sind Verbesserungen denkbar. Aus Bild 4.2.6 geht hervor, dass es reichen würde, die Muster zu speichern, welche die *Trennfläche* zwischen den Klassen bestimmen. Ein entsprechender Algorithmus ist in Bild 4.2.9 gezeigt. Dabei werden nur zwei Klassen Ω_1, Ω_2, repräsentiert durch zwei Stichproben ω_1, ω_2, angenommen; die Stichprobenumfänge seien N_1, N_2. Wie in Abschnitt 4.8.1 dargelegt wird, lässt sich ein Mehrklassenproblem stets auf mehrere Zweiklassenprobleme zurückführen.

Am Ende des Algorithmus in Bild 4.2.9 enthält K–GRENZE Paare von Mustern, die die Klassengrenze bestimmen. Auf die Muster in K–GRENZE kann noch der Algorithmus in Bild 4.2.7 angewendet werden. Allerdings ist der Algorithmus gemäß Bild 4.2.9 auf relativ klei-

Bilde eine zufällige Zerlegung der Stichprobe ω in L Teilmengen $\omega_1, \ldots, \omega_L$, $L \geq 3$.		
	FOR $i = 1$ TO L	
		$j = (i + 1) \bmod L$
		Klassifiziere Muster aus ω_i mit der NN Regel unter Verwendung von Mustern aus ω_j.
	Eliminiere alle Muster, die beim vorhergehenden Schritt falsch klassifiziert wurden.	
	Bilde aus den verbleibenden Mustern eine neue Stichprobe ω.	
UNTIL [in den letzten I Iterationen gab es keine Eliminationen mehr]		

Bild 4.2.10: Algorithmus zum Editieren einer Stichprobe

ne Stichproben beschränkt, da für jedes der $N_1 N_2$ Paare $\{ {}^i\boldsymbol{c}_1, {}^j\boldsymbol{c}_2\}$ der Abstand aller $N_1 + N_2$ Muster nach m_{ij} zu berechnen ist. Für $N_1 = N_2$ wächst also der Aufwand etwa mit N_1^3. Muster, die auf der Klassengrenze liegen, bilden auch die Basis für die Klassifikation mit Support Vektor Maschinen in Abschnitt 4.3; sie werden dort systematisch durch einen Optimierungsprozess gewonnen.

Eine Verbesserung der NN–Regel wird durch eine sog. *Editierung der Stichprobe* ω erreicht. Dafür gibt es verschiedene Ansätze, von denen einer als Beispiel in Bild 4.2.10 gezeigt ist. Er bildet relativ homogene Häufungsgebiete in der Stichprobe ω heraus. Der Effekt dieser Editierung ist, dass klar abgegrenzte Gebiete entstehen und Muster nahe den Klassengrenzen eliminiert werden. Am Ende ist ω die editierte Stichprobe.

Zwar gelten, wie schon erwähnt, alle Sätze über die NN– und mNN–Regel nur für einen Stichprobenumfang $N \to \infty$, jedoch wendet man diese natürlich stets auf endliche, oft sogar recht kleine, Stichproben an. Die Erwartung ist, dass Ergebnisse, die man für $N < \infty$ erhält, zumindest in ihrer Tendenz auch für $N \to \infty$ gelten. Die NN– oder mNN–Regel ist leicht realisierbar, der Rechenaufwand für kleinere Stichproben gering. Bei größeren Stichproben ist allerdings eine effiziente Suche nach den nächsten Nachbarn unverzichtbar; für Algorithmen dazu wird auf die Literatur verwiesen. Der NN–Klassifikator ist beispielsweise für eine schnelle Voruntersuchung zur Abschätzung der zu erwartenden Leistungsfähigkeit interessant. Es ist wichtig, für die Abstandsberechnung in (4.2.146) nur vergleichbare Merkmale zu verwenden, also solche mit gleicher Dimension bzw. dimensionslose; auf diesen Punkt wurde schon im Zusammenhang mit (2.5.47), S. 131, hingewiesen. Das Prinzip der Klassifikation nach dem nächsten Nachbarn wird auch bei der Suche in Multimedia Datenbanken verwendet.

4.3 Support Vektor Maschinen (VA.1.1.3, 13.04.2004)

Klassifikation mit sog. *Support Vektor Maschinen* (SVM) ist verglichen mit den statistischen Klassifikatoren ein noch relativ neuer Ansatz. Bei diesem wird eine für die Klassifikation wesentliche Teilmenge der Stichprobenelemente in der Trainingsmenge – eben die „Support Vektoren" – durch *konvexe quadratische Optimierung* in der Lernphase so bestimmt, dass die Klassentrennung im Merkmalsraum möglichst gut ist. Während insbesondere in Abschnitt 4.1 davon ausgegangen wurde, dass die erforderlichen statistischen Kenngrößen entweder exakt oder zumindest hinreichend genau bekannt sind, d. h. mit einer repräsentativen Stichprobe geschätzt wurden, wird hier ausdrücklich der Einfluss einer endlichen Stichprobe berücksichtigt. Die Basis dafür geht aus dem nächsten Abschnitt hervor. Bei der gesamten Diskussion wird nur ein Zweiklassenproblem betrachtet. Die beiden zu trennenden Klassen werden, wie in der Darstellung der SVM üblich, mit den Indizes $+1$ bzw. -1 versehen. Für mehr als zwei Klassen gibt es eine Reihe von Vorschlägen.

Ein k–Klassenproblem kann auf k Zweiklassenprobleme zurückgeführt werden, indem man k Klassifikatoren trainiert, die jeweils eine Klasse von den $k-1$ verbleibenden unterscheiden, die Strategie „eine gegen alle anderen". Der erste Klassifikator unterscheidet also die Klasse Ω_1 von den Klassen $\{\Omega_2, \ldots, \Omega_k\}$, der zweite die Klasse Ω_2 von den Klassen $\{\Omega_1, \Omega_3, \ldots, \Omega_k\}$, usw.

Ein weiterer Ansatz zur Rückführung des allgemeinen Klassifikationsproblems auf Zweiklassenprobleme ist charakterisiert durch die Kurzform „eine gegen eine". Dabei werden alle verschiedenen Paare von Klassen unterschieden, d. h. $k(k-1)/2$ Klassifikatoren zur Unterscheidung von ω_κ und ω_λ, $\kappa = 2, \ldots, k$, $\lambda = 1, \ldots, \kappa - 1$ realisiert. Für jede Klasse gibt es damit mehrere binäre Entscheidungen, sodass sich die Frage nach der endgültigen Entscheidung erhebt. Eine einfache und nach experimentellen Ergebnissen sehr wirksame Strategie besteht darin, bei jeder der durchgeführten Klassifikationen der dabei ausgewählten Klasse einen Punkt zu geben und sich dann endgültig für die Klasse mit maximaler Punktezahl zu entscheiden.

Obwohl hier mehr Klassifikatoren trainiert werden müssen als bei der Vorgehensweise „eine gegen alle anderen", ist der Rechenaufwand für Training und Test bei dieser Strategie wegen der kleineren Stichprobenumfänge geringer, soweit den in den Literaturhinweisen erwähnten Vergleichen zu entnehmen ist. Die Strategie „eine gegen eine" ist danach für das Mehrklassenproblem zu bevorzugen, zumal sie auch ausgezeichnete Erkennungsraten liefert.

Ein dritter Ansatz besteht darin, jeweils eine Klasse von den restlich noch verbleibenden zu unterscheiden, d. h. es werden $(k-1)$ Klassifikatoren zur Unterscheidung von ω_1 und $\{\omega_2, \omega_3, \ldots, \omega_k\}$, von ω_2 und $\{\omega_3, \omega_4, \ldots, \omega_k\}, \ldots$, und von ω_{k-1} und ω_k realisiert. Die Trennbarkeit der Klassen hängt hier i. Allg. von der gewählten Reihenfolge der Klassen ab.

Schließlich gibt es Ansätze, das Mehrklassenproblem, ähnlich wie bei den statistischen Klassifikatoren, „in einem Schritt" zu lösen. Da diese in einem experimentellen Vergleich nicht überzeugend abschnitten, wird dafür auf die Literatur verwiesen.

4.3.1 Die VC–Dimension

Von zwei zu trennenden Klassen sei eine klassifizierte Stichprobe von Mustern ${}^{\varrho}\boldsymbol{f}$, repräsentiert durch ihre Merkmale ${}^{\varrho}\boldsymbol{c} \in \mathbb{R}^n$, gegeben

$$\omega = \{\{{}^{\varrho}\boldsymbol{c}, y_\varrho\},\ \varrho = 1, 2, \ldots, N,\ y_\varrho \in \{-1, 1\}\}\ . \tag{4.3.1}$$

Die Merkmalsvektoren haben eine (unbekannte) Verteilungsfunktion $P(\boldsymbol{c}, y)$ und werden statistisch unabhängig aus einer Grundgesamtheit mit dieser Verteilung entnommen. Mit irgendeiner Menge T von Trennfunktionen $T = \{d_{\widetilde{\boldsymbol{a}}}(\boldsymbol{c})\}$, parametrisiert durch einen Parametervektor $\widetilde{\boldsymbol{a}}$, werden die Stichprobenelemente einer der beiden Klassen zugewiesen. Die mittleren Kosten bzw. das Risiko der Klassifikation wird definiert mit

$$V(d_{\widetilde{\boldsymbol{a}}}) = \int \frac{1}{2} \left| d_{\widetilde{\boldsymbol{a}}}({}^{\varrho}\boldsymbol{c}) - y \right| \, \mathrm{d}P(\boldsymbol{c}, y) \; . \tag{4.3.2}$$

Man vergleiche diese Definition mit der in (4.1.10), S. 309. Da die Verteilungsfunktion $P(\boldsymbol{c}, y)$ unbekannt ist, liegt es nahe, das Risiko durch das **empirische Risiko**

$$V_e(d_{\widetilde{\boldsymbol{a}}}) = \frac{1}{N} \sum_{\varrho=1}^{N} \frac{1}{2} \left| d_{\widetilde{\boldsymbol{a}}}({}^{\varrho}\boldsymbol{c}) - y_{\varrho} \right| \tag{4.3.3}$$

zu ersetzen. Bei endlichem, und insbesondere kleinem, Stichprobenumfang wird allerdings die Minimierung des empirischen Risikos i. Allg. *nicht* zu einer guten Klassifikationsleistung an einer neuen, von der Trainingsstichprobe disjunkten, Teststichprobe führen. Auf dieses Problem der *Generalisierung* wurde bereits in Abschnitt 1.3 hingewiesen.

Der Einfluss einer endlichen Stichprobe wird sich dadurch bemerkbar machen, dass das Risiko V *größer* ist als das empirische Risiko V_e. Der Unterschied ist durch folgende Abschätzung gegeben:

Satz 4.12 *Für jede Trennfunktion $d_{\widetilde{\boldsymbol{a}}}$ und jedes $N > h$ gilt mit der Wahrscheinlichkeit $1 - \eta$*

$$V(d_{\widetilde{\boldsymbol{a}}}) \;\leq\; V_e(d_{\widetilde{\boldsymbol{a}}}) + \phi\left(\frac{h}{N}, \frac{\log \eta}{N}\right) \; , \tag{4.3.4}$$

$$\phi\left(\frac{h}{N}, \frac{\log \eta}{N}\right) \;=\; \sqrt{\frac{h\left(\log\left(\frac{2N}{h}\right) + 1\right) - \log\left(\frac{\eta}{4}\right)}{N}} \; . \tag{4.3.5}$$

Dabei ist h die sog. VAPNIK–CHERVONENKIS-**Dimension** *(VC–Dimension).*
Beweis*: s. z. B. [Vapnik, 1995]*

Die obige Abschätzung ist *unabhängig* von der Verteilung $P(\boldsymbol{c}, y)$. Die linke Seite wird i. Allg. unbekannt bleiben, während die rechte bei bekanntem h für ein $d_{\widetilde{\boldsymbol{a}}}$ berechnet werden kann. Bei entsprechender Wahl von η, h, N kann $\phi > 1$ werden, d. h. die Abschätzung ist dann sicher nicht eng.

Die VC–Dimension h ist ein Maß für die **Kapazität** der Menge $\{d_{\widetilde{\boldsymbol{a}}} | \widetilde{\boldsymbol{a}} \in \mathbb{R}_{\widetilde{\boldsymbol{a}}}\}$ **von Trennfunktionen**. Ein Maß für die Kapazität eines Klassifikators gibt (4.10.5), S. 444. Für ein Zweiklassenproblem gibt h die maximale Zahl von Mustern an, die durch diese Funktionen in *alle* möglichen 2^h Partitionen zerlegt werden können. Diese Zerlegung muss nicht für alle Punktmengen vom Umfang h möglich sein, sondern für mindestens eine. Für jede mögliche Zerlegung gibt es also eine Trennfunktion, die diese korrekt durchführt. Eine spezielle Menge von Trennfunktionen ist die der *orientierten Hyperebenen*

$$d_{\widetilde{\boldsymbol{a}}}(\boldsymbol{c}) = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{a} + a_0 \; , \quad \text{mit} \quad \widetilde{\boldsymbol{a}} = \begin{pmatrix} a_0 \\ \boldsymbol{a} \end{pmatrix} \; , \tag{4.3.6}$$

mit denen für Punkte $\boldsymbol{c}$ entschieden werden kann, ob sie auf der positiven oder negativen Seite der Ebene liegen oder genau auf dieser Ebene. Die orientierten Ebenen $d_{\widetilde{\boldsymbol{a}}} = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{a} + a_0$ und

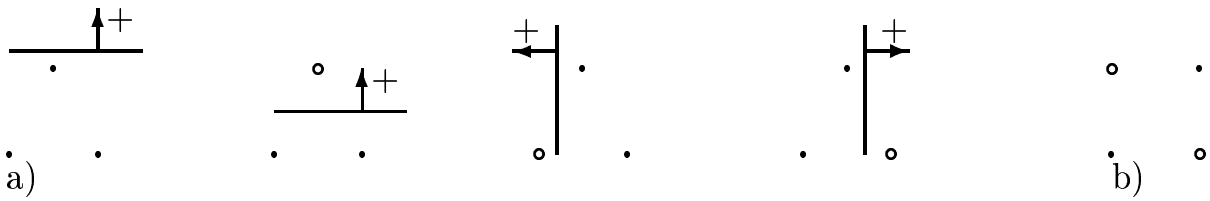


Bild 4.3.1: Zur VC–Dimension orientierter Geraden; a) drei Punkte in der Ebene lassen sich durch orientierte Geraden in alle acht Partitionen zerlegen; gezeigt sind vier Fälle, die anderen vier entstehen durch Vorzeichenumkehr; b) vier Punkte lassen sich nicht durch Geraden in alle $2^4 = 16$ Partitionen zerlegen

$d'_{\tilde{\boldsymbol{a}}} = -\boldsymbol{c}^{\mathsf{T}}\boldsymbol{a} - a_0$ sind *verschieden*, da sie Punkten, die nicht auf der Ebene liegen, gerade unterschiedliche Vorzeichen zuordnen.

Satz 4.13 *Die* VAPNIK–CHERVONENKIS*-Dimension der Menge orientierter Hyperebenen im* $\mathbb{R}^n$ *ist* $h = n + 1$.
Beweis*: s. z. B. [Burges, 1998]*

Bild 4.3.1 illustriert das in der Ebene. Es gibt drei Punkte, die sich durch eine geeignete orientierte Gerade in alle $2^3 = 8$ Partitionen zerlegen lassen, jedoch nicht vier Punkte – also ist $h = 3$, wie auch aus obigem Satz hervorgeht. Man wird erwarten, dass h umso größer ist, je mehr freie Parameter die Menge $\{d_{\tilde{\boldsymbol{a}}}\}$ hat. Hierzu gibt es jedoch Gegenbeispiele, die zeigen, dass dieses nicht i. Allg. zutrifft.

Der Term ϕ in (4.3.5) kennzeichnet den Unterschied zwischen dem tatsächlichen Risiko V und dem empirischen Risiko V_e und ist in dem Sinne ein Maß für das Konfidenzintervall des empirischen Risikos V_e, d. h. ϕ sollte *klein* sein. Das wird bei gegebenem Umfang N der Trainingsstichprobe ω durch eine *kleine* VC–Dimension h erreicht. Dagegen erwarten wir vom empirischen Risiko V_e, d. h. auch von der Fehlerrate auf der Trainingsstichprobe, dass es umso kleiner wird, je größer h wird. Von den beiden Termen in (4.3.4) nimmt also V_e mit wachsendem h ab, während ϕ mit wachsendem h zunimmt. Bei geeigneter Wahl von h sollte sich also ein minimaler Wert der rechten Seite von (4.3.4) ergeben, und damit eine optimale Generalisierung und eine minimale Fehlerrate auf einer neuen Teststichprobe. Dieser Beobachtung liegt die sog. *Minimierung des strukturellen Risikos* zugrunde, die unten betrachtet wird. Allerdings gilt i. Allg. *nicht*, dass eine *große* VC–Dimension notwendig eine *schlechte* Klassifikationsleistung zur Folge hat. Ein Gegenbeispiel dazu ist der nächste Nachbar Klassifikator in Abschnitt 4.2.7. Sein empirisches Risiko ist $V_e = 0$, seine VC-Dimension $h = \infty$. Trotzdem arbeitet er erfahrungsgemäß sehr gut.

Die Minimierung des strukturellen Risikos geht von (4.3.4) aus. Der Konfidenzterm ϕ hängt über h von der gewählten *Menge* $\{d_{\tilde{\boldsymbol{a}}}\}$ von Trennfunktionen ab. Der Term V_e hängt von der speziell durch *Training* aus $\{d_{\tilde{\boldsymbol{a}}}\}$ bestimmten Funktion ab. Man geht z. B. so vor, dass man eine „Struktur“ vorprägt, indem man eine Menge T von Trennfunktionen, z. B. Polynome in $\boldsymbol{c}$ vom Grade q, in geschachtelte Teilmengen $T_1 \subset T_2 \subset \ldots \subset T_n$ mit $h_1 < h_2 < \ldots < h_n$ zerlegt. Für jedes T_i trainiert man einen Klassifikator, der das empirische Risiko minimiert. Dann wählt man unter den n Klassifikatoren den aus, der die Abschätzung des Risikos in (4.3.4) bzw. die Fehlerrate auf einer disjunkten Teststichprobe minimiert. Von diesem kann man erwarten, dass er auch eine gute Generalisierungsleistung hat, d. h. eine geringe Fehlerrate auf einer neuen

Teststichprobe aufweist.

Definition 4.10 *Die* **Support Vektor Maschine** *arbeitet so, dass die Abschätzung des empirischen Risikos in (4.3.4) minimiert wird.*

Man vergleiche diese Definition mit der in Definition 4.1, S. 305. Dort wird das Risiko minimiert; das erfordert vollständige statistische Information, wie in (4.1.1), S. 306, vorausgesetzt, bzw. eine im Sinne von (1.3.1), S. 19, repräsentative, d. h. genügend große Stichprobe. Die Abschätzung des Risikos in (4.3.4) berücksichtigt den Einfluss einer kleinen Stichprobe im Term ϕ. Dieser geht mit wachsendem Stichprobenumfang gegen Null, sodass beide Definitionen ineinander übergehen.

4.3.2 Linear separierbare Stichprobe

Wir betrachten zunächst ein Zweiklassenproblem und nehmen an, dass die Klassen mit einer Hyperebene *exakt* trennbar sind; dieses ist die sog. **lineare Separierbarkeit**. Von den beiden Klassen sei eine klassifizierte Stichprobe gemäß (4.3.1) gegeben. Nach Voraussetzung liegt damit eine *linear separierbare Stichprobe* vor. Im Folgenden werden Muster ${}^{\varrho}\boldsymbol{f}$ mit $y_\varrho = 1$ bzw. mit $y_\varrho = -1$ als „positive Stichprobenelemente“ bzw. „negative Stichprobenelemente“ bezeichnet. Wenn eine Hyperebene mit Parametern $\widetilde{\boldsymbol{a}} = (a_0, a_1, \ldots, a_n)^\mathsf{T}$ gegeben ist, die die positiven exakt von den negativen Stichprobenelementen trennt, gelte für alle Muster aus der Stichprobe

$$
\begin{aligned}
{}^{\varrho}\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 &\geq +1, \quad \text{wenn } y_\varrho = +1\,, && (4.3.7)\\
{}^{\varrho}\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 &\leq -1, \quad \text{wenn } y_\varrho = -1\,, && (4.3.8)\\
y_\varrho\left({}^{\varrho}\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0\right) &\geq 1\,, \quad \forall\, {}^{\varrho}\boldsymbol{c} \in \omega\,. && (4.3.9)
\end{aligned}
$$

Es wird daran erinnert, dass für die Hyperebene in (4.3.6) die Beziehungen

$$
\begin{aligned}
\boldsymbol{n} &= \frac{-\boldsymbol{a}}{\sqrt{\boldsymbol{a}^\mathsf{T}\boldsymbol{a}}} = \frac{-\boldsymbol{a}}{|\boldsymbol{a}|}\,, \qquad \text{(Normalenvektor mit Betrag Eins)}, && (4.3.10)\\
s_0 &= \frac{-a_0}{|\boldsymbol{a}|}\,, \qquad \text{(Ebenenabstand vom Ursprung)}, && (4.3.11)\\
s_{\boldsymbol{c}} &= \frac{-(\boldsymbol{a}^\mathsf{T}\boldsymbol{c} + a_0)}{|\boldsymbol{a}|}\,, \qquad \text{(Ebenenabstand von Punkt } \boldsymbol{c}\text{)}, && (4.3.12)
\end{aligned}
$$

gelten. Ein Abstand von der Hyperebene ist positiv, wenn der Punkt in Richtung der Normalen von der Ebene abliegt. Offensichtlich kann man die Ebenenparameter so normieren, dass für den Punkt $\boldsymbol{c}'$, der der Hyperebene am nächsten liegt, $|\boldsymbol{c}'^\mathsf{T}\boldsymbol{a} + a_0| = 1$ gilt, da die Gleichungen $\boldsymbol{a}^\mathsf{T}\boldsymbol{c} + a_0 = 0$ bzw. $\gamma(\boldsymbol{a}^\mathsf{T}\boldsymbol{c} + a_0) = 0$ die gleichen Ebenen definieren. Die Beziehungen (4.3.7) und (4.3.8) entsprechen also einer geeigneten Skalierung des Vektors $\widetilde{\boldsymbol{a}}$.

Wir betrachten nun positive bzw. negative Stichprobenelemente aus ω, für die in (4.3.7) bzw. (4.3.8) das Gleichheitszeichen gilt. Sie liegen also auf den Hyperebenen ${}^{\varrho}\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 = 1$ bzw. ${}^{\varrho}\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 = -1$. Die Ebenen haben beide den Normalenvektor $\boldsymbol{a}$, d. h. sie sind *parallel*. In Bild 4.3.2 sind dies die beiden parallelen gestrichelten Linien. Die Beträge ihrer senkrechten Abstände vom Ursprung sind $|1-a_0|/|\boldsymbol{a}|$ bzw. $|-1-a_0|/|\boldsymbol{a}|$. Die beiden Ebenen haben also voneinander den Abstand $2/|\boldsymbol{a}|$. Zwischen den beiden Ebenen liegen *keine* Stichprobenelemente, da lineare Separierbarkeit vorausgesetzt wurde. Ein sinnvolles Optimierungskriterium zur Wahl

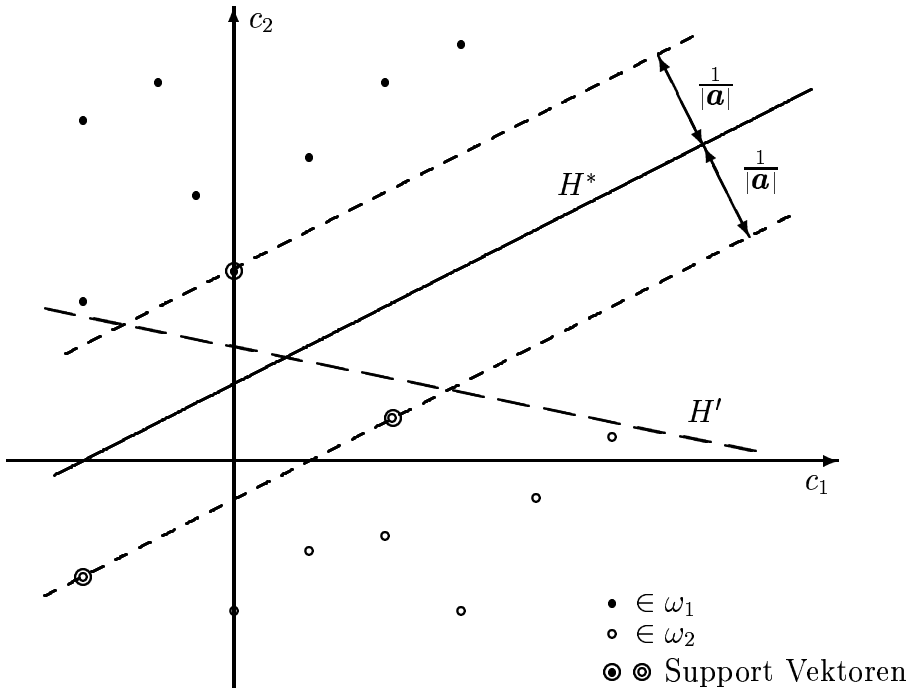


Bild 4.3.2: Trennung zweier linear separierbarer Stichproben durch eine optimale Hyperebene H^* und eine schlechte H'

der Trennebenen ist daher die *Maximierung* des Abstandes der beiden Ebenen bzw. die *Minimierung* von $|\boldsymbol{a}|^2$ unter der Nebenbedingung von (4.3.9). Dieses ergibt die optimale Trennebene H^* in Bild 4.3.2. Natürlich kann eine linear separierbare Stichprobe i. Allg. mit vielen anderen Ebenen getrennt werden, wie z. B. H' in Bild 4.3.2, jedoch sind diese offenbar fehleranfälliger bei der Klassifikation einer neuen Teststichprobe. Diejenigen Vektoren der Trainingsstichprobe, die *genau* auf der Hyperebene H^* liegen, also für die in (4.3.9) das Gleichheitszeichen gilt, sind die *Support Vektoren*. Aus Bild 4.3.2 geht anschaulich hervor, dass sie hinreichend sind, die optimale Trennebene zu definieren, d. h. alle anderen Elemente der Stichprobe sind in diesem Sinne unwesentlich. Jede Änderung eines oder mehrerer Support Vektoren würde andererseits die optimale Lösung verändern, d. h. sie sind notwendig.

Definition 4.11 *Der* optimale lineare Klassifikator *für eine linear separierbare Stichprobe von Mustern aus zwei Klassen ist gegeben durch die Hyperebene*

$$H^* : \; d_{\tilde{\boldsymbol{a}}} = {}^{\varrho}\boldsymbol{c}^{\mathsf{T}}\boldsymbol{a} + a_0 = 0 \,, \tag{4.3.13}$$

mit der Minimierungsbedingung

$$\frac{1}{2}|\boldsymbol{a}|^2 = \min \tag{4.3.14}$$

und den N Nebenbedingungen

$$y_{\varrho}\left({}^{\varrho}\boldsymbol{c}^{\mathsf{T}}\boldsymbol{a} + a_0\right) - 1 \geq 0 \,, \quad \forall\, {}^{\varrho}\boldsymbol{c} \in \omega \,. \tag{4.3.15}$$

Die Entscheidungsregel ist

$$\boldsymbol{c} \in \begin{cases} \Omega_1 & : \; d_{\tilde{\boldsymbol{a}}} \geq 0 \\ \Omega_2 & : \; d_{\tilde{\boldsymbol{a}}} < 0 \end{cases} \tag{4.3.16}$$

Diese Definition eines optimalen Klassfikators vergleiche man mit der in (4.1.33), S. 314. Dort wurde als Optimierungskriterium die Minimierung der mittleren Kosten verwendet, hier die Maximierung des Abstandes der beiden oben eingeführten Ebenen.

4.3.3 Zur Lösung des Optimierungsproblems

Für die Lösung des in Definition 4.11 eingeführten Optimierungsproblems (4.3.14) mit den N Nebenbedingungen (4.3.15) werden positive LAGRANGE-Multiplikatoren $\boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_N)^\mathsf{T}$ eingeführt (s. Abschnitt 1.6). Dadurch werden die Nebenbedingungen reduziert auf Nebenbedingungen für die LAGRANGE-Multiplikatoren, die einfacher zu handhaben sind. Bei Nebenbedingungen $n_i > 0$ werden diese mit *positiven* LAGRANGE-Multiplikatoren multipliziert und von der Optimierungsbedingung *subtrahiert*. Damit ergibt sich die LAGRANGE-Gleichung

$$L(\widetilde{\boldsymbol{a}},\, \boldsymbol{\vartheta}) = \frac{1}{2}|\boldsymbol{a}|^2 - \sum_{\varrho=1}^{N} \vartheta_\varrho \left(y_\varrho \left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 \right) - 1 \right) \;, \tag{4.3.17}$$

die bezüglich $\boldsymbol{a}, a_0$ zu *minimieren* ist mit der Nebenbedingung, dass die Ableitungen nach $\boldsymbol{\vartheta}$ verschwinden. Diese Aufgabe ist als *konvexe quadratische Optimierung* bekannt. Wegen Einzelheiten zu ihrer Lösung wird auf die in Abschnitt 4.11 zitierte Literatur verwiesen.

Die KARUSH–KUHN–TUCKER-Bedingungen (s. Satz 1.3, S. 38) für die Optimierung von L sind

$$\frac{\partial L}{\partial \boldsymbol{a}} = \boldsymbol{a} - \sum_{\varrho=1}^{N} \vartheta_\varrho y_\varrho {}^\varrho\boldsymbol{c} = 0 \;, \tag{4.3.18}$$

$$\frac{\partial L}{\partial a_0} = -\sum_{\varrho=1}^{N} \vartheta_\varrho y_\varrho = 0 \;, \tag{4.3.19}$$

$$0 \le \vartheta_\varrho \;, \quad \varrho = 1, \ldots, N \tag{4.3.20}$$

$$0 \le y_\varrho \left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 \right) - 1 \;, \quad \varrho = 1, \ldots, N \tag{4.3.21}$$

$$0 = \vartheta_\varrho \left(y_\varrho \left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 \right) - 1 \right) \;, \quad \varrho = 1, \ldots, N \;. \tag{4.3.22}$$

Sie sind notwendig und hinreichend für die Lösung des Optimierungsproblems.

Statt des primären Problems (4.3.17) kann auch ein *duales Problem* betrachtet werden, insbesondere wenn dieses einfacher lösbar ist; dieses ist die übliche Vorgehensweise bei SVM. Das duale Problem besteht darin, (4.3.17) zu *maximieren* mit der Nebenbedingung, dass die Ableitungen nach $\widetilde{\boldsymbol{a}}$ verschwinden. Diese Ableitungen aus (4.3.18) und (4.3.19) in (4.3.17) eingesetzt ergeben das *duale Problem*

$$\boxed{\begin{aligned} L_d(\boldsymbol{\vartheta}) &= \sum_{\varrho=1}^{N} \vartheta_\varrho - \frac{1}{2} \sum_{\varrho=1}^{N} \sum_{\sigma=1}^{N} \vartheta_\varrho \vartheta_\sigma y_\varrho y_\sigma \left({}^\varrho\boldsymbol{c}^{\mathsf{T}\,\sigma}\boldsymbol{c} \right) \;, \\ 0 &\le \vartheta_\varrho \;, \\ 0 &= \sum_{\varrho=1}^{N} \vartheta_\varrho y_\varrho \;. \end{aligned}} \tag{4.3.23}$$

Das Training einer SVM erfordert also die Maximierung von L_d bezüglich $\boldsymbol{\vartheta}$ unter den Nebenbedingungen $\vartheta_\varrho \geq 0,\ \sum_\varrho \vartheta_\varrho y_\varrho = 0$. Die Parameter $\boldsymbol{a}$ folgen dann aus (4.3.18).

Setzt man (4.3.18) in (4.3.13) ein, so erhält man für die optimale Trennebene

$$H^* :\ d_{\widetilde{\boldsymbol{a}}} = \sum_{\varrho=1}^{N} \vartheta_\varrho y_\varrho \left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{c}\right) + a_0 = 0\ . \tag{4.3.24}$$

Das bedeutet, dass man zur Berechnung der Trennebene *nur die Skalarprodukte* von Merkmalsvektoren braucht. Dieses ist die Basis für den Übergang von Trennebenen auf praktisch beliebige Polynome in den Koeffizienten des Merkmalsvektors in Abschnitt 4.3.5; dabei werden Kernfunktionen wie in (3.8.50) – (3.8.52), S. 234, genutzt, um mit (3.8.53), S. 234, Skalarprodukte hochdimensionaler Vektoren über niedrigdimensionale zu berechnen.

Aus den KARUSH–KUHN–TUCKER-Bedingungen (4.3.21), (4.3.22) geht hervor, dass diese entweder durch Punkte (Merkmalsvektoren) *genau* auf der Hyperebene und LAGRANGE-Multiplikatoren $\vartheta_\varrho = 0$ aber auch $\vartheta_\varrho > 0$ oder durch Punkte *nicht* auf der Hyperebene und LAGRANGE-Multiplikatoren $\vartheta_\varrho = 0$ erfüllt werden können. Aus (4.3.24) sieht man, dass in die Berechnung der optimalen Hyperebene *nur* solche Merkmalsvektoren mit *aktiven Nebenbedingungen* $\vartheta_\varrho > 0$ eingehen; alle anderen Merkmalsvektoren sind unerheblich.

Definition 4.12 *Die* **Support Vektoren** *einer Stichprobe sind die Merkmalsvektoren, die in die Berechnung der optimalen Hyperebene in (4.3.24) eingehen; das sind die Merkmalsvektoren mit* aktiven Nebenbedingungen *bzw. mit* LAGRANGE*-Multiplikatoren* $\vartheta_\varrho > 0$.

4.3.4 Linear nicht separierbare Stichprobe

In konkreten Anwendungen werden Stichproben i. Allg. *nicht* linear separierbar sein. Trotzdem lässt sich die in Abschnitt 4.3.2 vorgestellte Vorgehensweise mit geeigneten Modifikationen auch hier verfolgen. Das Prinzip besteht darin, die Nebenbedingungen (4.3.7) und (4.3.8) durch zusätzliche *Schlupfvariable* ("slack variables") $\xi_\varrho \geq 0$ dann abzuschwächen, wenn dieses wegen der Durchdringung der Stichproben notwendig wird. Die Nebenbedingungen (4.3.7) – (4.3.9) werden also ersetzt durch

$$\begin{aligned}
{}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 &\geq +1 - \xi_\varrho, \quad \text{wenn } y_\varrho = +1\ , && (4.3.25)\\
{}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 &\leq -1 + \xi_\varrho, \quad \text{wenn } y_\varrho = -1\ , && (4.3.26)\\
y_\varrho\left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0\right) &\geq 1 - \xi_\varrho\ , \quad \varrho = 1, \ldots, N\ , && (4.3.27)\\
\xi_\varrho &\geq 0\ , \quad \varrho = 1, \ldots, N\ . && (4.3.28)
\end{aligned}$$

Aufgrund der Diskussion in Abschnitt 4.3.2 kann es auf der Trainingsstichprobe nur dann einen Fehler geben, wenn $\xi_\varrho > 1$ wird. Eine Abschätzung dieser Fehler ist also $\sum_\varrho \xi_\varrho$. Statt wie in (4.3.14) $\frac{1}{2}|\boldsymbol{a}|^2$ zu minimieren, ist es daher sinnvoll $\frac{1}{2}|\boldsymbol{a}|^2 + \gamma\sum_\varrho \xi_\varrho$ zu minimieren. Mit zusätzlichen LAGRANGE-Multiplikatoren β_ϱ wird die Nichtnegativität der Schlupfvariablen erreicht. Damit ergibt sich nun die LAGRANGE-Gleichung

$$L(\widetilde{\boldsymbol{a}},\ \boldsymbol{\vartheta}) = \frac{1}{2}|\boldsymbol{a}|^2 + \gamma\sum_{\varrho=1}^{N}\xi_\varrho - \sum_{\varrho=1}^{N}\vartheta_\varrho\left(y_\varrho\left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0\right) - 1 + \xi_\varrho\right) - \sum_{\varrho=1}^{N}\beta_\varrho\xi_\varrho\ . \tag{4.3.29}$$

Die KARUSH–KUHN–TUCKER-Bedingungen dafür sind (4.3.18) – (4.3.20) sowie *zusätzlich*

$$0 \ \leq\ y_\varrho\left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0\right) - 1 + \xi_\varrho\ , \tag{4.3.30}$$

$$
\begin{aligned}
0 &= \vartheta_\varrho \left(y_\varrho \left({}^\varrho\boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0 \right) - 1 + \xi_\varrho \right) \ , && (4.3.31)\\
\frac{\partial L}{\partial \xi_\varrho} &= \gamma - \vartheta_\varrho - \beta_\varrho = 0 \ , && (4.3.32)\\
0 &\leq \xi_\varrho \ , && (4.3.33)\\
0 &\leq \beta_\varrho \ , && (4.3.34)\\
0 &= \beta_\varrho \xi_\varrho \ , && (4.3.35)\\
&\quad \text{jeweils für } \ \varrho = 1, \ldots, N \ .
\end{aligned}
$$

Auch hier ist das duale Problem nützlich; es lautet

$$
\begin{aligned}
L_d(\boldsymbol{\vartheta}) &= \sum_{\varrho=1}^{N} \vartheta_\varrho - \frac{1}{2} \sum_{\varrho=1}^{N} \sum_{\sigma=1}^{N} \vartheta_\varrho \vartheta_\sigma y_\varrho y_\sigma \left({}^\varrho\boldsymbol{c}^{\mathsf{T}\sigma}\boldsymbol{c} \right) \ , && (4.3.36)\\
0 &\leq \vartheta_\varrho \leq \gamma \ , && (4.3.37)\\
0 &= \sum_{\varrho=1}^{N} \vartheta_\varrho y_\varrho \ . && (4.3.38)
\end{aligned}
$$

Es stimmt also mit (4.3.23) überein, nur die Nebenbedingungen $\vartheta_\varrho \leq \gamma$ kommen hinzu. Der Gewichtsvektor der Hyperebene ist mit (4.3.18)

$$
\boldsymbol{a} = \sum_{\varrho=1}^{N} \vartheta_\varrho y_\varrho {}^\varrho\boldsymbol{c} \ . \tag{4.3.39}
$$

Die Konstante a_0 erhält man aus (4.3.31) und (4.3.35). Die optimale Hyperebene hat wie im Falle der linearen Separierbarkeit die Form

$$
\boxed{H^* : \ d_{\widetilde{\boldsymbol{a}}} = \sum_{\varrho=1}^{N} \vartheta_\varrho y_\varrho \left(\boldsymbol{c}^{\mathsf{T}\,\varrho}\boldsymbol{c} \right) + a_0 = 0 \ ,} \tag{4.3.40}
$$

d. h. auch zu ihrer Berechnung sind nur Skalarprodukte der Merkmalsvektoren erforderlich. Natürlich wird man die Summe nur über die Support Vektoren erstrecken. Zur numerischen Berechnung der Support Vektoren wird auf die Anmerkungen in Abschnitt 4.11 verwiesen. Die Klassifikation erfolgt also nicht mit dem Parametervektor $\boldsymbol{a}$ und der Ebenengleichung (4.3.6) sondern mit der Entwicklung (4.3.39) des Parametervektors und (4.3.40). Die Entscheidungsregel ist (4.3.16).

4.3.5 Nichtlineare Trennfunktionen

Die bisherige Beschränkung auf Hyperebenen als Trennfunktionen lässt sich auf relativ einfache Weise wesentlich verallgemeinern. Der Schlüssel dafür ist die Beobachtung, dass sowohl beim Training, nämlich in (4.3.36) – (4.3.38), als auch bei der Klassifikation, nämlich in (4.3.40), *nur Skalarprodukte* der Merkmalsvektoren berechnet werden müssen.

Wenn man den Merkmalsvektor $\boldsymbol{c} \in \mathbb{R}^n$ mit einer Abbildung $\boldsymbol{\phi}$ in einen höherdimensionalen Raum $\widetilde{\boldsymbol{c}} \in \mathbb{R}^{\widetilde{n}}$, $n < \widetilde{n}$, transformiert, so können alle obigen Rechnungen in (4.3.25) – (4.3.40) statt mit $\boldsymbol{c}$ nun mit $\widetilde{\boldsymbol{c}}$ durchgeführt werden; d. h. auch von dem neuen Merkmalsvektor $\widetilde{\boldsymbol{c}}$ müssen bei Training und Klassifikation *nur Skalarprodukte* $\widetilde{\boldsymbol{c}}^\mathsf{T}\widetilde{\boldsymbol{c}}$ berechnet werden. Statt einer

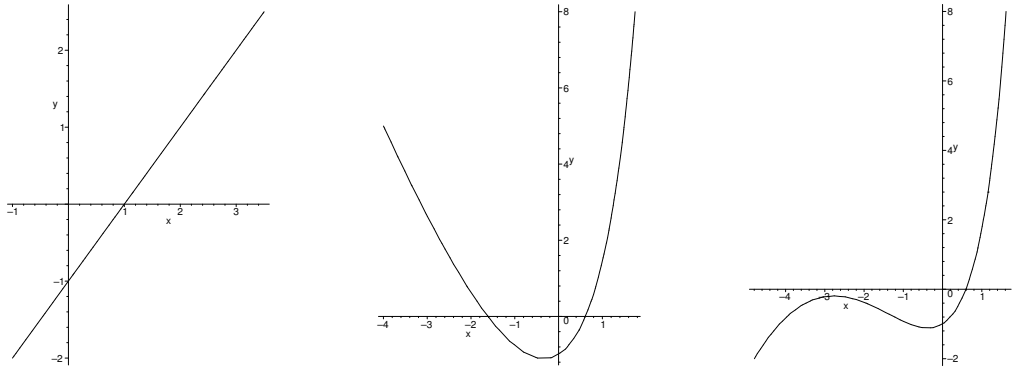


Bild 4.3.3: von links nach rechts: Beispiele für Trennfunktionen (Polynome) (4.3.44) vom Grad 1, 2 und 3

Trennebene $d = \boldsymbol{c}^\mathsf{T}\boldsymbol{a} + a_0$ hat man dann eine *nichtlineare Trennfunktion* $d' = \boldsymbol{\phi}(\boldsymbol{c})^\mathsf{T}\boldsymbol{a}' + a'_0$. Beispiele für resultierende Trennfunktionen im $\mathbb{R}^1$ zeigt Bild 4.3.3.

In Abschnitt 3.8.3, (3.8.49) – (3.8.53), S. 234, wurde gezeigt, dass man die Berechnung des Skalarprodukts in einem hochdimensionalen Raum immer dann vermeiden kann, wenn es eine geeignete Kernfunktion gibt mit der Eigenschaft

$$K({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \boldsymbol{\phi}({}^i\boldsymbol{c})^\mathsf{T}\boldsymbol{\phi}({}^j\boldsymbol{c}) \ . \tag{4.3.41}$$

Beispiele für solche Kernfunktionen wurden in Abschnitt 3.8.3 angegeben. Das duale Optimierungsproblem in (4.3.36) lässt sich mit der Kernfunktion kompakt angeben zu

$$L_d(\boldsymbol{\vartheta}) = \boldsymbol{e}^\mathsf{T}\boldsymbol{\vartheta} - \frac{1}{2}\boldsymbol{\vartheta}^\mathsf{T}\boldsymbol{Q}\boldsymbol{\vartheta} \tag{4.3.42}$$

sowie den Nebenbedingungen (4.3.37) und (4.3.38). Dabei ist $\boldsymbol{e}$ ein Vektor, dessen Komponenten alle Eins sind, und $\boldsymbol{Q}$ ist eine Matrix mit Elementen $q_{ij} = y_i y_j K({}^i\boldsymbol{c}, {}^j\boldsymbol{c})$.

Wenn beim Training einer SVM N_s Support Vektoren ${}^\varrho\boldsymbol{c}_\mathrm{s}$ berechnet wurden, ist der Gewichtsvektor in Verallgemeinerung von (4.3.39)

$$\boldsymbol{a} = \sum_{\varrho=1}^{N_\mathrm{s}} \vartheta_\varrho y_\varrho \boldsymbol{\phi}({}^\varrho\boldsymbol{c}_\mathrm{s}) \ , \tag{4.3.43}$$

und die Klassifikation erfolgt in Verallgemeinerung von (4.3.40) mit der nichtlinearen Trennfunktion

$$\boxed{T^* : \ d_{\tilde{\boldsymbol{a}}} \ = \ \sum_{\varrho=1}^{N_\mathrm{s}} \vartheta_\varrho y_\varrho K(\boldsymbol{c}, {}^\varrho\boldsymbol{c}_\mathrm{s}) + a_0 \ .} \tag{4.3.44}$$

Die Rechenkomplexität ist damit für die Kernfunktionen (3.8.50) – (3.8.52) praktisch genauso groß wie im linearen Fall (4.3.40). Klassifikatoren auf dieser Basis haben erfahrungsgemäß eine ausgezeichnete Leistung.

Zur Verwendung normierter Kernfunktionen der Form

$$K_\mathrm{n}({}^i\boldsymbol{c}, {}^j\boldsymbol{c}) = \frac{K({}^i\boldsymbol{c}, {}^j\boldsymbol{c})}{\sqrt{K({}^i\boldsymbol{c}, {}^i\boldsymbol{c})K({}^j\boldsymbol{c}, {}^j\boldsymbol{c})}} \tag{4.3.45}$$

wird auf die zitierte Literatur verwiesen.

4.4 Polynomklassifikator (VA.2.2.3, 07.09.2005)

4.4.1 Annahmen

Wegen der in Abschnitt 4.2 erwähnten Probleme bei der Ermittlung von bedingten Verteilungsdichten ist es sinnvoll, Ansätze für die Klassifikation von Mustern zu untersuchen, die ohne die Schätzung der Verteilungsdichte auskommen. Ein solcher Ansatz besteht darin, k Trennfunktionen $d_\lambda(\boldsymbol{c})$ einzuführen, welche eine Klassifikation gemäß der Bedingung

$$\text{wenn } d_\kappa(\boldsymbol{c}) = \max_\lambda d_\lambda(\boldsymbol{c}) \text{ , dann entscheide } \boldsymbol{c} \in \Omega_\kappa \tag{4.4.1}$$

erlauben. Man kann diese als direkte Verallgemeinerung von (4.1.33) bzw. (4.1.34), S. 315, auffassen, wo die $d_\lambda(\boldsymbol{c})$ den Termen $p_\lambda p(\boldsymbol{c}|\Omega_\lambda)$ bzw. $p(\Omega_\lambda|\boldsymbol{c})$ entsprechen. Analog zu (4.2.123) ist die Trennfläche zwischen zwei Klassen Ω_κ und Ω_λ durch $d_\kappa(\boldsymbol{c}) = d_\lambda(\boldsymbol{c})$ gegeben. Das Problem besteht nun darin, geeignete Funktionen $d_\lambda(\boldsymbol{c})$ *ohne Rückgriff* auf bedingte Dichten zu bestimmen. Die wesentliche Annahme dabei ist, dass die **Trennfunktionen** $d_\lambda(\boldsymbol{c})$ Elemente einer vorgegebenen parametrischen Familie $\widetilde{d}$ von Funktionen sind, d. h. es gilt

$$d_\lambda(\boldsymbol{c}) = d(\boldsymbol{c}, \boldsymbol{a}_\lambda) \in \widetilde{d}(\boldsymbol{c}, \boldsymbol{a}) = \{d(\boldsymbol{c}, \boldsymbol{a}) \,|\, \boldsymbol{a} \in \mathbb{R}_{\boldsymbol{a}}\} \; . \tag{4.4.2}$$

Dabei ist $\boldsymbol{a}$ ein Parametervektor und $\mathbb{R}_{\boldsymbol{a}}$ der Parameterraum. Auf den ersten Blick scheint vielleicht (4.4.2) keinen Fortschritt gegenüber (4.1.1) zu bringen, jedoch ist nicht vorausgesetzt, dass die Funktionen $d(\boldsymbol{c}, \boldsymbol{a})$ Verteilungsdichten sind. Die Parameter $\boldsymbol{a}_\lambda$ sind so zu wählen, dass (4.4.1) für möglichst viele Muster $\boldsymbol{c} \in \Omega$ zu einer richtigen Entscheidung führt. Für eine einfache mathematische Behandlung ist es zweckmäßig, die Familie $\widetilde{d}$ auf Funktionen einzuschränken, die *linear in den Parametern* $\boldsymbol{a}$ sind. Diese spezielle Familie sei

$$\widetilde{d}_1(\boldsymbol{c}, \boldsymbol{a}) \;=\; \left\{\boldsymbol{a}^\mathsf{T}\boldsymbol{\varphi}(\boldsymbol{c}) \,|\, \boldsymbol{a} \in \mathbb{R}_{\boldsymbol{a}},\; \varphi_\nu(\boldsymbol{c}),\; \nu = 1, \ldots, \mathrm{m}\right\} \; , \tag{4.4.3}$$

wobei die $\varphi_\nu(\boldsymbol{c})$ *linear unabhängige Funktionen* sind. Mit dieser zweiten Annahme wird die Menge der Funktionen zwar eingeschränkt, jedoch lässt sich im Prinzip eine Funktion, die stetig ist und n-te Ableitungen besitzt, durch eine TAYLOR-Reihe, also eine Funktion aus (4.4.3), approximieren. Das einfachste Beispiel einer Familie $\widetilde{d}_1$ sind die in c_ν linearen Funktionen oder *Hyperebenen* (s. (4.3.13), S. 364), bei denen

$$\boldsymbol{\varphi}(\boldsymbol{c}) = (1, c_1, c_2, \ldots, c_n)^\mathsf{T} \tag{4.4.4}$$

und $\boldsymbol{a}$ ein $(n+1)$-dimensionaler Vektor ist. Eine Verallgemeinerung ergeben die quadratischen Funktionen bzw. Polynome in n Variablen vom Grad zwei

$$\begin{aligned} \boldsymbol{\varphi}(\boldsymbol{c}) \;&=\; (1, c_1, c_2, \ldots, c_n, c_1c_1, c_2c_1, \ldots, c_nc_n)^\mathsf{T} \\ &=\; \left(1, \boldsymbol{\varphi}_{\mathrm{rest}}(\boldsymbol{c})^\mathsf{T}\right)^\mathsf{T} \; , \end{aligned} \tag{4.4.5}$$

bei denen $\boldsymbol{a}$ ein $(1 + n + n(n+1)/2)$-dimensionaler Vektor ist. Grundsätzlich ist es möglich, für $\boldsymbol{\varphi}(\boldsymbol{c})$ *Polynome* beliebiger Ordnung zu verwenden, und daher wird der resultierende Klassifikator auch als **Polynomklassifikator** bezeichnet . Da ein Polynom p-ter Ordnung in n Variablen $c_\nu, \nu = 1, \ldots, n$ aber $\binom{n+p}{p}$ Koeffizienten hat (s. (3.8.43), S. 232), beschränkt man sich praktisch meistens auf die Ordnung $p = 2$ oder $p = 3$.

Der Ansatz (4.4.1) zusammen mit (4.4.3), (4.4.5) ergibt einen Klassifikator, der in seiner Struktur mit Bild 4.2.5 identisch ist. Die Methoden zur Bestimmung der Parameter, nämlich $\widetilde{\boldsymbol{a}}$

in (4.2.122), $\boldsymbol{a}$ in (4.3.13) und $\boldsymbol{a}$ in (4.4.3), sind jedoch verschieden, sodass die resultierenden Klassifikatoren ebenfalls verschieden sind.

Außer dem Polynomansatz kann man für $\boldsymbol{\varphi}(\boldsymbol{c})$ auch stückweise lineare Funktionen verwenden. Die in Abschnitt 4.4.3 als „direkte“ Lösung bezeichnete Vorgehensweise ist dann jedoch nicht möglich. Man muss iterative Lern- oder Trainingsalgorithmen verwenden, für die auf die Literatur verwiesen wird.

4.4.2 Optimierungsaufgabe

Gemäß (4.4.1) – (4.4.3) haben die Trennfunktionen $d_\lambda(\boldsymbol{c})$ die spezielle Form

$$d_\lambda(\boldsymbol{c}) = \boldsymbol{a}_\lambda^{\mathsf{T}} \boldsymbol{\varphi}(\boldsymbol{c}) \;, \qquad \lambda = 1, \ldots, k \;, \tag{4.4.6}$$

wobei die Funktionen $\varphi_\nu(\boldsymbol{c})$ des Vektors

$$\boldsymbol{\varphi}(\boldsymbol{c}) = (\varphi_1(\boldsymbol{c}), \ldots, \varphi_m(\boldsymbol{c}))^{\mathsf{T}} \tag{4.4.7}$$

bekannte, linear unabhängige Funktionen sind. Wie erwähnt, sind die Parametervektoren $\boldsymbol{a}_\lambda$ so zu bestimmen, dass (4.4.1) für möglichst viele Muster $\boldsymbol{c} \in \Omega$ zu einer richtigen Entscheidung führt. Im Abschnitt 4.1 wurde die Bestimmung der Entscheidungsregel $\delta(\Omega_\lambda \,|\, \boldsymbol{c})$ auf die Optimierungsaufgabe (4.1.11) zurückgeführt; analog wird auch hier die Bestimmung der Parameter $\boldsymbol{a}_\lambda$ auf eine Optimierungsaufgabe zurückgeführt. Die Verwendung des Risikos (4.1.12) ist hier nicht möglich, da dieses Kenntnisse über bedingte Dichten erfordert. Statt dessen wird der Ansatz verwendet, mit Hilfe der Trennfunktionen $d_\lambda(\boldsymbol{c})$ eine vorgegebene *ideale Trennfunktion* $\delta_\lambda(\boldsymbol{c})$ möglichst gut zu approximieren. Eine mögliche Wahl der idealen Trennfunktion ist in Analogie zu (4.1.33)

$$\text{wenn } \boldsymbol{c} \in \Omega_\kappa \;, \text{ dann } \delta_\kappa(\boldsymbol{c}) = 1 \;, \text{ sonst } \delta_\lambda(\boldsymbol{c}) = 0 \;, \text{ für } \lambda \neq \kappa \;. \tag{4.4.8}$$

Wenn (4.4.8) gilt und die Funktionen d_λ eine fehlerfreie Approximation von δ_λ sind, werden offensichtlich alle Muster aus dem Problemkreis Ω richtig klassifiziert. Im Allgemeinen wird jedoch $d_\lambda \neq \delta_\lambda$ sein, sodass Fehlklassifikationen möglich sind. Als Kriterium für die Güte der Approximation von δ_λ durch d_λ wird der mittlere quadratische Fehler

$$\varepsilon = E\left\{ \sum_{\lambda=1}^{k} (\delta_\lambda(\boldsymbol{c}) - d_\lambda(\boldsymbol{c}))^2 \right\} \tag{4.4.9}$$

gewählt. Gesucht werden Parameter $\boldsymbol{a}_\lambda$, sodass ε minimiert wird.

Um die Parameter konkret zu berechnen, ist es zweckmäßig, den Fehler ε noch etwas kompakter anzugeben. Die k Trennfunktionen d_λ werden in dem Vektor

$$\boldsymbol{d}(\boldsymbol{c}) = (d_1(\boldsymbol{c}), \ldots, d_k(\boldsymbol{c}))^{\mathsf{T}} \tag{4.4.10}$$

zusammengefasst, ebenso die k idealen Trennfunktionen δ_λ in dem Vektor

$$\boldsymbol{\delta}(\boldsymbol{c}) = (\delta_1(\boldsymbol{c}), \ldots, \delta_k(\boldsymbol{c}))^{\mathsf{T}} \;. \tag{4.4.11}$$

Gemäß (4.4.8) ist $\boldsymbol{\delta}(\boldsymbol{c})$ also ein Vektor, bei dem für ein Muster $\boldsymbol{c} \in \Omega_\kappa$ sämtliche Komponenten den Wert Null haben mit Ausnahme der κ-ten Komponente, die den Wert Eins hat. Mit einer Parametermatrix

$$\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k) \tag{4.4.12}$$

gilt schließlich

$$\begin{aligned} \boldsymbol{d}(\boldsymbol{c}) &= \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi}(\boldsymbol{c}) \,, & (4.4.13)\\ \varepsilon(\boldsymbol{A}) &= E\left\{(\boldsymbol{\delta}(\boldsymbol{c}) - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi}(c))^2\right\} \,. & (4.4.14) \end{aligned}$$

Die gesuchte optimale Parametermatrix $\boldsymbol{A}^*$ ist definiert durch

$$\boldsymbol{A}^* = \underset{\boldsymbol{A}}{\operatorname{argmin}}\, \varepsilon(\boldsymbol{A}) \,. \qquad (4.4.15)$$

Definition 4.13 *Der Polynomklassifikator minimiert den Erwartungswert des mittleren quadratischen Fehlers (4.4.14), indem die ideale Trennfunktion (4.4.11) durch Trennfunktionen (4.4.13) approximiert wird; die Funktionen $\boldsymbol{\varphi}(\boldsymbol{c})$ sind dabei Polynome in c_ν.*

Statt des quadratischen Abstandes zwischen d_λ und δ_λ in (4.4.9) kann im Prinzip irgendein anderer – z. B. $|d_\lambda - \delta_\lambda|$ verwendet werden. Ebenso ist es möglich, andere ideale Trennfunktionen als die in (4.4.8) zu wählen – z. B. $\delta_\kappa = 1$ wenn $\boldsymbol{c} \in \Omega_\kappa$ und $\delta_\lambda = -1$ für $\lambda \neq \kappa$. Schließlich gibt es verschiedene Möglichkeiten zur Wahl der Funktionen $\varphi_\nu(\boldsymbol{c})$ – zwei Beispiele sind in (4.4.4), (4.4.5) angegeben, die stückweise linearen Trennfunkionen wurden erwähnt – und verschiedene Vorgehensweisen zur Berechnung der Parameter $\boldsymbol{a}_\lambda$. Wir beschränken uns hier auf den obigen Ansatz wegen seiner mathematischen Einfachheit, seiner noch zu erörternden Beziehung zu statistischen Klassifikatoren und seiner hohen Leistungsfähigkeit.

4.4.3 Berechnung der Trennfunktionen

Wie aus der obigen Diskussion hervorgeht reduziert sich bei dem gewählten Ansatz die Berechnung der Trennfunktionen auf die Berechnung der unbekannten Parameter. Dafür werden im Folgenden mehrere Vorgehensweisen genannt und eine genauer erörtert. Zuvor wird aber noch kurz auf die allgemeine Trennfunktion, ohne Spezialisierung auf (4.4.3), eingegangen.

1. Regression

Mit (4.4.10), (4.4.11), aber ohne die spezielle Form (4.4.3), ist der Fehler

$$\varepsilon = E\left\{(\boldsymbol{\delta}(\boldsymbol{c}) - \boldsymbol{d}(\boldsymbol{c}))^2\right\} \qquad (4.4.16)$$

bezüglich $\boldsymbol{d}(\boldsymbol{c})$ zu minimieren. Die Lösung ist ein grundlegendes Ergebnis aus der Theorie der Schätzungen bzw. der Regressionsanalyse.

Satz 4.14 *Die Funktion $\boldsymbol{d}^*(\boldsymbol{c})$, die (4.4.16) minimiert, ist der* bedingte Erwartungswert *von $\boldsymbol{\delta}$, wenn $\boldsymbol{c}$ beobachtet wurde; dieser ist*

$$\boldsymbol{d}^*(\boldsymbol{c}) = E\{\boldsymbol{\delta}\,|\,\boldsymbol{c}\} \,. \qquad (4.4.17)$$

Beweis*: s. z. B. [Rao, 1973, Papageorgiou, 1991].*

Der bedingte Erwartungswert in (4.4.17) ist definitionsgemäß

$$\begin{aligned} \boldsymbol{d}^* &= E\{\boldsymbol{\delta}\,|\,\boldsymbol{c}\}\\ &= \sum_{\lambda=1}^{k} p(\Omega_\lambda\,|\,\boldsymbol{c})\boldsymbol{\delta}(\boldsymbol{c}) \end{aligned}$$

$$= \quad (p(\Omega_1|\boldsymbol{c}), \ldots, p(\Omega_k|\boldsymbol{c}))^{\mathsf{T}} \ . \tag{4.4.18}$$

Die letzte Zeile ergibt sich aus der speziellen Wahl von $\boldsymbol{\delta}$ gemäß (4.4.8), (4.4.11). Die beste Trennfunktion $\boldsymbol{d}^*(\boldsymbol{c})$, die für die gewählte ideale Trennfunktion $\boldsymbol{\delta}(\boldsymbol{c})$ in (4.4.8) den mittleren quadratischen Fehler minimiert, ist also der Vektor der *a posteriori Wahrscheinlichkeiten* (4.4.18). Ein Vergleich von (4.1.33), (4.1.34) und (4.4.1), (4.4.18) ergibt, dass dieser Klassifikator mit dem optimalen BAYES-Klassifikator identisch ist. Dieses Ergebnis wird zusammengefasst in

Satz 4.15 *Der* BAYES-Klassifikator *(4.1.33), der die Fehlerwahrscheinlichkeit minimiert, und der* Polynomklassifikator *(4.4.1), der bei uneingeschränkter Trennfunktion die mittlere quadratische Abweichung von der idealen Trennfunktion (4.4.8) minimiert, sind identisch.*

Der obige Satz liefert eine theoretische Begründung dafür, dass der gewählte Ansatz gegenüber anderen möglichen vorgezogen wird. Die Funktion $\boldsymbol{d}^*(\boldsymbol{c})$ in (4.4.17) wird als **Regressionsfunktion** bezeichnet. Die einfache geschlossene Form von (4.4.17) darf nicht darüber hinwegtäuschen, dass die Berechnung von $\boldsymbol{d}^*$ i. Allg. keineswegs einfach ist. Aus (4.4.18) und (4.1.34) geht hervor, dass dafür *vollständige* statistische Information, insbesondere die bedingten Dichten $p(\boldsymbol{c}|\Omega_\kappa)$, erforderlich ist. Wenn man jedoch die zulässigen Funktionen $\boldsymbol{d}(\boldsymbol{c})$ wie in (4.4.3) einschränkt, ergeben sich numerisch auswertbare Gleichungen für die Berechnung der Trennfunktionen, wie im Folgenden gezeigt wird. Bei den so eingeschränkten Trennfunktionen ist, wie erwähnt, nur noch die Parametermatrix $\boldsymbol{A}$ zu berechnen. Dafür eignen sich folgende Verfahren.

2. Direkte Lösung

Gesucht ist die Matrix $\boldsymbol{A}^*$, welche den Fehler (4.4.14)

$$\varepsilon \quad = \quad E\left\{(\boldsymbol{\delta}(\boldsymbol{c}) - \boldsymbol{A}^{\mathsf{T}}\varphi(\boldsymbol{c}))^2\right\}$$

minimiert. Als direkte Lösung bezeichnen wir hier die Auswertung der bekannten Bedingung, dass dann die partiellen Ableitungen von ε nach den Elementen a_{ij} von $\boldsymbol{A}$ verschwinden müssen. Das Ergebnis ist

Satz 4.16 *Die optimale Parametermatrix $\boldsymbol{A}^*$, die (4.4.14) minimiert, erhält man aus*

$$\boldsymbol{A}^* = \left[E\left\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\varphi}^{\mathsf{T}}(\boldsymbol{c})\right\}\right]^{-1} E\left\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\delta}^{\mathsf{T}}(\boldsymbol{c})\right\} \ . \tag{4.4.19}$$

Dabei ist die Existenz der inversen Matrix von $E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathsf{T}}\}$ vorausgesetzt.

Beweis: Notwendige Bedingung für ein relatives Extremum ist

$$\begin{aligned}
\frac{\partial \varepsilon}{\partial \boldsymbol{A}} \quad &= \quad \left(\frac{\partial \varepsilon}{\partial a_{ij}}\right) = \boldsymbol{0} \ , \\
\frac{\partial \varepsilon}{\partial a_{ij}} \quad &= \quad \frac{E\left\{\partial(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})^2\right\}}{\partial a_{ij}} \\
&= \quad E\left\{2\left(\delta_j - \sum_\nu a_{\nu j}\varphi_j\right)(-\varphi_i)\right\} \ ,
\end{aligned} \tag{4.4.20}$$

$$\frac{\partial \varepsilon}{\partial \boldsymbol{A}} \;=\; 2E\left\{\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{A} - \boldsymbol{\varphi}\boldsymbol{\delta}^{\mathsf{T}}\right\} = \mathbf{0}\;. \tag{4.4.21}$$

Die Einzelheiten der obigen Ableitung erhält man einfach, wenn man den Fehler ε mit Hilfe der Elemente a_{ij} und Komponenten δ_μ, φ_ν ausdrückt. Ein Vergleich von (4.4.21) mit dem Beweis von Satz 3.1, S. 167, in Abschnitt 3.2.1 zeigt, dass auch hier eine Orthogonalitätsbedingung vorliegt. Wegen der Wahl der Funktionen $\boldsymbol{\varphi}$ in (4.4.5) ist die erste Komponente eine Eins. Daraus und aus (4.4.21) folgt, dass der Erwartungswert des Fehlers verschwindet, nämlich

$$\begin{aligned}
\mathbf{0} \;&=\; E\left\{\boldsymbol{\varphi}(\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{A} - \boldsymbol{\delta}^{\mathsf{T}})\right\} \\
&=\; E\left\{\begin{pmatrix} 1 \\ \boldsymbol{\varphi}_{\text{rest}} \end{pmatrix}(\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{A} - \boldsymbol{\delta}^{\mathsf{T}})\right\} \;=\; E\left\{\begin{pmatrix} (\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi} - \boldsymbol{\delta})^{\mathsf{T}} \\ \boldsymbol{\varphi}_{\text{rest}}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi} - \boldsymbol{\delta})^{\mathsf{T}} \end{pmatrix}\right\}\;, \\
\mathbf{0} \;&=\; E\left\{\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{A} - \boldsymbol{\delta}^{\mathsf{T}}\right\} \;=\; E\left\{\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi} - \boldsymbol{\delta}\right\}\;.
\end{aligned} \tag{4.4.22}$$

Die Berechnung von $\boldsymbol{A}^*$ in (4.4.19) ist mit einer klassifizierten Stichprobe möglich. Bekanntlich gilt für den **Erwartungswert** einer Funktion $g(x)$

$$E\{g(x)\} = \int g(x)p(x)\,\mathrm{d}x \simeq \frac{1}{N}\sum_{j=1}^{N} g({}^j x)\;, \tag{4.4.23}$$

sodass sich die Erwartungswerte in (4.4.19) schätzen lassen mit

$$\begin{aligned}
E\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\varphi}^{\mathsf{T}}(\boldsymbol{c})\} \;&\simeq\; \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\varphi}({}^j\boldsymbol{c})\boldsymbol{\varphi}^{\mathsf{T}}({}^j\boldsymbol{c})\;, \\
E\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\delta}^{\mathsf{T}}(\boldsymbol{c})\} \;&\simeq\; \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\varphi}({}^j\boldsymbol{c})\boldsymbol{\delta}^{\mathsf{T}}({}^j\boldsymbol{c})\;.
\end{aligned} \tag{4.4.24}$$

Es wird daran erinnert, dass für alle ${}^j\boldsymbol{c} \in \omega$ der Wert von $\boldsymbol{\delta}({}^j\boldsymbol{c})$ bekannt ist, wenn ω eine klassifizierte Stichprobe ist.

Definiert man eine Matrix $\boldsymbol{\Phi}$, deren N Spalten die m-dimensionalen Merkmalsvektoren der Stichprobe enthalten, sowie eine Matrix $\boldsymbol{\Delta}$, deren N Spalten die k-dimensionalen Einheitsvektoren der Klassenzugehörigkeiten enthalten, so lässt sich (4.4.19) mit den Schätzwerten in (4.4.24) angeben; die beiden Matrizen sind

$$\boldsymbol{\Phi} \;=\; \begin{pmatrix} \varphi_1({}^1\boldsymbol{c}) & \varphi_1({}^2\boldsymbol{c}) & \dots & \varphi_1({}^N\boldsymbol{c}) \\ \varphi_2({}^1\boldsymbol{c}) & \varphi_2({}^2\boldsymbol{c}) & \dots & \varphi_2({}^N\boldsymbol{c}) \\ \vdots & \vdots & & \vdots \\ \varphi_m({}^1\boldsymbol{c}) & \varphi_m({}^2\boldsymbol{c}) & \dots & \varphi_m({}^N\boldsymbol{c}) \end{pmatrix}\;, \tag{4.4.25}$$

$$\boldsymbol{\Delta} \;=\; \begin{pmatrix} \delta_1({}^1\boldsymbol{c}) & \delta_1({}^2\boldsymbol{c}) & \dots & \delta_1({}^N\boldsymbol{c}) \\ \delta_2({}^1\boldsymbol{c}) & \delta_2({}^2\boldsymbol{c}) & \dots & \delta_2({}^N\boldsymbol{c}) \\ \vdots & \vdots & & \vdots \\ \delta_k({}^1\boldsymbol{c}) & \delta_k({}^2\boldsymbol{c}) & \dots & \delta_k({}^N\boldsymbol{c}) \end{pmatrix}\;. \tag{4.4.26}$$

Damit ist z. B.

$$E\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\varphi}^{\mathsf{T}}(\boldsymbol{c})\} \simeq \frac{1}{N}\boldsymbol{\Phi}\,\boldsymbol{\Phi}^{\mathsf{T}}\;, \tag{4.4.27}$$

und (4.4.19) geht über in

$$\boxed{\boldsymbol{A}^* = \left(\frac{1}{N}\boldsymbol{\Phi}\,\boldsymbol{\Phi}^\mathsf{T}\right)^{-1}\left(\frac{1}{N}\boldsymbol{\Phi}\boldsymbol{\Delta}^\mathsf{T}\right) \;=\; \boldsymbol{B}^{-1}\boldsymbol{D}\;.} \tag{4.4.28}$$

Wenn der Vektor $\boldsymbol{\varphi}$ von Funktionen φ_ν aus m Komponenten besteht und k Klassen zu unterscheiden sind, ist $\boldsymbol{A}^*$ eine $m \times k$ Matrix, zu deren Berechnung die Inversion einer $m \times m$ Matrix erforderlich ist. Für einen Merkmalvektor mit n Komponenten c_ν und ein vollständiges Polynom p-ten Grades in c_ν ist wie in (3.8.43), S. 232,

$$m = \binom{n+p}{p} = \frac{(n+p)(n+p-1)\ldots(n+1)}{1\cdot 2\cdot \ldots \cdot p}\;. \tag{4.4.29}$$

Daraus ergibt sich, dass die Inversion von $E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^\mathsf{T}\}$ schon für p = 2 ab etwa $n = 100$, entsprechend m = 5151, ein Problem wird. Dazu kommt ein weiteres Problem. Wegen den in der Regel zwischen Merkmalen auftretenden linearen Abhängigkeiten ist es bei Wahl der φ_ν gemäß (4.4.4), (4.4.5) möglich, dass die Matrix $E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^\mathsf{T}\}$ nicht regulär ist. Aus diesen Gründen ist es zweckmäßig, die Berechnung von $\boldsymbol{A}$ wie in Abschnitt 4.4.4 erläutert vorzunehmen. Die dort beschriebene Vorgehensweise erlaubt es zudem, in jedem Rechenschritt genau den Term des Polynoms hinzu zu nehmen, der zur stärksten Abnahme des Approximationsfehlers ε in (4.4.14) führt. Nach ν Schritten erhält man die beste Lösung mit den ν besten Merkmalen bzw. den besten Termen des Polynoms, d. h. man hat eine *analytische Merkmalsauswahl*; man kann dieses auch als *sparsamen Schätzwert* des Regressionspolynoms bezeichnen, da alle nicht genügend guten Terme unterdrückt werden.

3. Iterative Lösung

Die in (4.4.15) geforderte Minimierung von (4.4.14) ist grundsätzlich auch iterativ mit dem Ansatz

$$\boldsymbol{A}_{N+1} = \boldsymbol{A}_N - \beta_N \boldsymbol{R}_N \tag{4.4.30}$$

möglich, wobei $\boldsymbol{A}_0$ eine beliebige Anfangsmatrix, $\boldsymbol{R}_N$ die Richtung im N-ten Iterationsschritt und β_N ein Faktor ist, der die Schrittweite bestimmt (s. (1.6.7), S. 36). Als Richtungen kommen der Gradient $\partial\varepsilon/\partial\boldsymbol{A}$ in (4.4.21) oder auch ein zyklisches Durchlaufen aller Koordinaten, in diesem Falle aller Elemente von $\boldsymbol{A}$, in Frage.

4. Stochastische Approximation

Beim iterativen Ansatz (4.4.30) muss der in (4.4.14) bzw. in (4.4.21) auftretende Erwartungswert geschätzt werden. Das ist möglich, wenn eine klassifizierte Stichprobe bekannt ist. Wenn aber Muster laufend beobachtet werden und bei jeder neuen Beobachtung ein verbesserter Wert für $\boldsymbol{A}$ berechnet werden soll, eignet sich (4.4.30) nicht. Das Problem kann prinzipiell mit der *stochastischen Approximation* (s. (1.6.29), S. 39) gelöst werden. Zu minimieren sei i. Allg. eine Funktion

$$g(\boldsymbol{A}) = E\left\{s(\boldsymbol{A}, \boldsymbol{c})\right\}\;. \tag{4.4.31}$$

wobei s von den Parametern $\boldsymbol{A}$ und der Zufallsvariablen $\boldsymbol{c}$ abhängt; ein Spezialfall ist (4.4.14). Ausgehend von einem beliebigen Startwert $\boldsymbol{A}_0$ wird bei Beobachtung des N–ten Wertes ${}^N\boldsymbol{c}$ der Zufallsvariablen $\boldsymbol{c}$ ein verbesserter Wert

$$\boldsymbol{A}_N = \boldsymbol{A}_{N-1} - \beta_N \nabla_{\boldsymbol{A}} s(\boldsymbol{A}_{N-1}, {}^N\boldsymbol{c}) \tag{4.4.32}$$

berechnet. Obwohl die zu minimierende Funktion $g(\boldsymbol{A})$ einen Erwartungswert enthält, ist dessen Kenntnis in (4.4.32) nicht erforderlich. Bedingungen für die Konvergenz der stochastischen Approximation (4.4.32) sind in der erwähnten Literatur angegeben.

5. Rekursive Berechnung

Bezeichnet man in (4.4.23) den mit N Stichprobenelementen geschätzten Erwartungswert mit $E\{g\}_N$, so gilt

$$\begin{aligned} E\{g(\boldsymbol{c})\} &\simeq \frac{1}{N}\sum_{\varrho=1}^{N} g({}^\varrho\boldsymbol{c}) = E\{g\}_N \,, \\ E\{g\}_N &= \frac{1}{N}\left(\sum_{\varrho=1}^{N-1} g({}^\varrho\boldsymbol{c}) + g({}^N\boldsymbol{c})\right) \\ &= \frac{N-1}{N} E\{g\}_{N-1} + \frac{1}{N} g({}^N\boldsymbol{c}) \,. \end{aligned} \tag{4.4.33}$$

Der Schätzwert $E\{g\}_N$ kann also mit dem vorherigen Schätzwert $E\{g\}_{N-1}$ der neuen Beobachtung ${}^N\boldsymbol{c}$ berechnet werden. Eine verallgemeinerte Form ist

$$E\{g\}_N = (1-\beta_N) E\{g\}_{N-1} + \beta_N g({}^N\boldsymbol{c}) \,, \tag{4.4.34}$$

wobei in (4.4.33) $\beta_N = N^{-1}$ ist. Entsprechendes gilt auch für (4.4.14). Setzt man in (4.4.19)

$$\boldsymbol{A}_N = \left[E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^\mathsf{T}\}_N\right]^{-1} E\{\boldsymbol{\varphi}\boldsymbol{\delta}^\mathsf{T}\}_N \tag{4.4.35}$$

und führt die mit N Stichprobenwerten berechneten Erwartungswerte entsprechend (4.4.33) auf die mit $(N-1)$ Werten berechneten zurück, ergibt sich eine Beziehung zwischen $\boldsymbol{A}_N$ und $\boldsymbol{A}_{N-1}$. Die Rechnung ergibt

$$\boldsymbol{A}_N = \boldsymbol{A}_{N-1} + \beta_N \left[E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^\mathsf{T}\}_N\right]^{-1} \boldsymbol{\varphi}({}^N\boldsymbol{c}) \left(\boldsymbol{\delta}^\mathsf{T}({}^N\boldsymbol{c}) - \boldsymbol{\varphi}^\mathsf{T}({}^N\boldsymbol{c}) \boldsymbol{A}_{N-1}\right) \,. \tag{4.4.36}$$

Der Bezug zwischen (4.4.32) und (4.4.36) ist offensichtlich, da in diesem Falle

$$-\nabla_{\boldsymbol{A}} s = \boldsymbol{\varphi}(\boldsymbol{\delta}^\mathsf{T} - \boldsymbol{\varphi}^\mathsf{T}\boldsymbol{A}) \tag{4.4.37}$$

ist. Eine Verallgemeinerung ist die zusätzliche Matrix $[E\{\boldsymbol{\varphi}\boldsymbol{\varphi}\}_N]^{-1}$ in (4.4.36). Approximiert man diese durch die Einheitsmatrix (“quick and dirty”), sind (4.4.32) und (4.4.36) identisch.

Gleichungen zur Parameterberechnung wie in (4.4.32) oder in (4.4.36) spielen bei Lernalgorithmen eine große Rolle, da sie die laufende Verbesserung des Klassifikators aufgrund neuer Beobachtungen ermöglichen. Darauf wird im Abschnitt 4.8 nochmals eingegangen.

4.4.4 Zur numerischen Berechnung der Parametermatrix

Wie bereits in Abschnitt 4.4.3 erwähnt wurde, ist es nicht zweckmäßig, die optimale Parametermatrix in Satz 4.16 durch Inversion der Matrix $E\left\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\varphi}^{\mathsf{T}}(\boldsymbol{c})\right\}$ zu berechnen, sondern diese iterativ mit dem GAUSS–JORDAN-Algorithmus zu ermitteln; dessen Grundzüge werden daher zunächst kurz vorgestellt.

Prinzip des GAUSS–JORDAN-Algorithmus

Gegeben seien quadratische Matrizen $\boldsymbol{E}$, $\boldsymbol{F}$, $\boldsymbol{G}$. Wenn für das Produkt

$$\boldsymbol{E}\,\boldsymbol{F} = \boldsymbol{G} \tag{4.4.38}$$

gilt, dass $\boldsymbol{G}$ eine Einheitsmatrix ist, so ist $\boldsymbol{F}$ definitionsgemäß die inverse Matrix zu $\boldsymbol{E}$. Ein Element g_{ij} der Matrix $\boldsymbol{G}$ erhält man aus

$$g_{ij} = \sum_{k=1}^{N} e_{ik} f_{kj} \ . \tag{4.4.39}$$

Aus dieser Gleichung folgen drei Beobachtungen, die die drei wesentlichen Operationen des Verfahrens ergeben:

1. Die Elemente einer Zeile von $\boldsymbol{F}$ ändern sich nicht, wenn man jedes Element der korrespondierenden Zeilen von $\boldsymbol{E}$ und $\boldsymbol{G}$ mit einer Konstanten r multipliziert. Wählt man in der i-ten Zeile $r = 1/e_{ii}$, so wird das entsprechende Hauptdiagonalelement auf Eins normiert.
2. Die Elemente einer Zeile von $\boldsymbol{F}$ ändern sich nicht, wenn man zu der Zeile i in den Matrizen $\boldsymbol{E}$ und $\boldsymbol{G}$ die jeweilige Zeile k aus diesen Matrizen addiert. Damit lassen sich alle Elemente einer Spalte der Matrix $\boldsymbol{E}$ zu Null machen.
3. Die Elemente von $\boldsymbol{G}$ ändern sich nicht, d.h. (4.4.39) bleibt unverändert, wenn man die Spalten j_1 und j_2 von $\boldsymbol{E}$ vertauscht und gleichzeitig die Zeilen j_1 und j_2 von $\boldsymbol{F}$. Damit kann man sicherstellen, dass in der Hauptdiagonale immer das größte Element aus der aktuellen Zeile steht, dass also in Schritt 1 die Division durch eine sehr kleine Zahl oder gar durch Null vermieden wird.

Die obigen Schritte 1 und 2 sind bereits hinreichend, um die Matrix $\boldsymbol{E}$ sukzessive in eine Einheitsmatrix $\boldsymbol{I}$ zu transformieren a). Der Schritt 3 wird als **Pivotisierung** bezeichnet und empfiehlt sich stets; er dient der numerischen Stabilität b). Statt der Pivotisierung nach dem maximalen Zeilenelement sind aus Sicht der Musterklassifikation auch andere Strategien denkbar. Da, wie erwähnt, $\boldsymbol{G}$ mit einer Einheitsmatrix initialisiert wurde, $\boldsymbol{F}$ in diesem Falle die inverse von $\boldsymbol{E}$ ist, d. h. $\boldsymbol{F} = \boldsymbol{E}^{-1}$, und $\boldsymbol{F}$ durch die obigen Umformungen nicht verändert wurde, enthält nun die umgeformte Matrix $\boldsymbol{G}'$ die gesuchte inverse Matrix von $\boldsymbol{E}$, da die Ausgangsgleichung (4.4.38) $\boldsymbol{E}\,\boldsymbol{F} = \boldsymbol{I}$ umgeformt wurde zu $\boldsymbol{I}\,\boldsymbol{F} = \boldsymbol{G}'$.

Die schrittweise Umformung der Matrix $\boldsymbol{E}$ in die Einheitsmatrix $\boldsymbol{I}$ wird auch als *Normierung* bezeichnet. Die j-te Spalte von $\boldsymbol{E}$ heißt normiert, wenn sie eine Eins in der j-ten Komponente enthält und sonst Nullen.

Für die Auswahl der zu normierenden Spalte, d. h. die Bestimmung des Pivotelementes zum Zwecke der Berechnung der optimalen Parametermatrix $\boldsymbol{A}^*$ des Polynomklassifikators, gibt es folgende Kriterien:

1. Die Auswahl des maximalen Elements in der aktuellen Zeile, wie in Schritt 3 oben erwähnt.
2. Die Auswahl des Polynomterms, der minimale lineare Abhängigkeit zu den schon ausgewählten hat.
3. Die Auswahl des Polynomterms, der zur maximalen Verminderung des Approximationsfehlers ε führt.

Die beiden letzten Kriterien sowie die Anwendung des GAUSS–JORDAN-Algorithmus zur Berechnung von $\boldsymbol{A}^*$ werden im Folgenden genauer betrachtet.

Minimierung der linearen Abhängigkeiten

Nach (4.4.28) gilt für die optimale Parametermatrix $\boldsymbol{A}^* = \boldsymbol{B}^{-1}\boldsymbol{D}$. Mit den in (4.4.25) – (4.4.28) eingeführten Bezeichnungen wird eine $m \times (m+k)$ Matrix

$$(\boldsymbol{B}\,|\,\boldsymbol{D}) = \left(\frac{1}{N}\boldsymbol{\Phi}\,\boldsymbol{\Phi}^{\mathsf{T}}\,\middle|\,\frac{1}{N}\boldsymbol{\Phi}\boldsymbol{\Delta}^{\mathsf{T}}\right) \;\approx\; \left(E\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\varphi}^{\mathsf{T}}(\boldsymbol{c})\}\,|\,E\{\boldsymbol{\varphi}(\boldsymbol{c})\boldsymbol{\delta}^{\mathsf{T}}(\boldsymbol{c})\}\right) \tag{4.4.40}$$

eingeführt. Diese wird von links mit $\boldsymbol{B}^{-1}$ multipliziert und ergibt

$$\boldsymbol{B}^{-1}(\boldsymbol{B}\,|\,\boldsymbol{D}) = (\boldsymbol{I}\,|\,\boldsymbol{B}^{-1}\boldsymbol{D}) \;=\; (\boldsymbol{I}\,|\,\boldsymbol{A}^*)\;. \tag{4.4.41}$$

Die Normierung der linken Spalten der Ausgangsmatrix $(\boldsymbol{B}\,|\,\boldsymbol{D})$ ist genau das, was der oben beschriebene GAUSS–JORDAN-Algorithmus bewirkt .

Eine geeignete Pivotisierung empfiehlt sich auch für diesen Fall. Man kann zeigen, dass die Pivotisierung durch Auswahl des größten Hauptdiagonalelementes der Auswahl desjenigen Merkmals (bzw. desjenigen Polynomterms) entspricht, das am wenigsten linear abhängig von den schon ausgewählten Merkmalen ist. Die schon ausgewählten Merkmale entsprechen bereits normierten Spalten der Matrix $\boldsymbol{B}$. Für die Qualität der Merkmale ist es sinnvoll, solche zu wählen, die keine oder nur geringe lineare Abhängigkeiten aufweisen, wie es auch schon in Abschnitt 3.9.2 diskutiert wurde. Bricht man die Normierung nach $m' < m$ Schritten ab, d. h. normiert nur m' Spalten der Matrix $\boldsymbol{B}$, so erhält man die optimale Parametermatrix $\boldsymbol{A}^{*'}$ mit m' Zeilen und k Spalten. Das bedeutet, dass man statt der m Polynomterme, welche die Merkmale für die Klassifikation darstellen, nur die m' mit den geringsten linearen Abhängigkeiten verwendet. Man bekommt also durch das Verfahren eine *analytische Merkmalsauswahl*.

Maximierung der Abnahme des Fehlers

Statt der Pivotisierung durch Auswahl des größten Hauptdiagonalelementes ist es aus Sicht der Klassifikation sinnvoll, diejenige Spalte zu normieren, die zur größten Abnahme des Fehlers ε in (4.4.14) führt. Zu diesem Zweck wird die Matrix in (4.4.40) erweitert zu der Matrix

$$\begin{aligned}
\boldsymbol{S} &= E\{(\boldsymbol{\varphi}^{\mathsf{T}},\boldsymbol{\delta}^{\mathsf{T}})^{\mathsf{T}}(\boldsymbol{\varphi}^{\mathsf{T}},\boldsymbol{\delta}^{\mathsf{T}})\} \\
&= \begin{pmatrix} E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathsf{T}}\}^{m\times m} & E\{\boldsymbol{\varphi}\boldsymbol{\delta}^{\mathsf{T}}\}^{m\times k} \\ E\{\boldsymbol{\delta}\boldsymbol{\varphi}^{\mathsf{T}}\}^{k\times m} & E\{\boldsymbol{\delta}\boldsymbol{\delta}^{\mathsf{T}}\}^{k\times k} \end{pmatrix}^{(m+k)\times(m+k)} \\
&\approx \frac{1}{N}\begin{pmatrix} \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathsf{T}} & \boldsymbol{\Phi}\boldsymbol{\Delta}^{\mathsf{T}} \\ \boldsymbol{\Delta}\boldsymbol{\Phi}^{\mathsf{T}} & \boldsymbol{\Delta}\boldsymbol{\Delta}^{\mathsf{T}} \end{pmatrix}\;.
\end{aligned} \tag{4.4.42}$$

Wegen der Definition der idealen Trennfunktionen $\boldsymbol{\delta}$ in (4.4.8) und (4.4.11) ist $E\{\boldsymbol{\delta}\boldsymbol{\delta}^{\mathsf{T}}\}$ eine Diagonalmatrix $\mathbf{diag}(p_\kappa)$, deren Hauptdiagonalelemente die a priori Wahrscheinlichkeiten p_κ der k Klassen sind. Wenn man den Spaltenvektor der Trennfunktionen gemäß (4.4.5) in der Form $\boldsymbol{\varphi}(\boldsymbol{c}) = (1, \boldsymbol{\varphi}_{\text{rest}}^{\mathsf{T}})^{\mathsf{T}}$ verwendet und im Vektor $\boldsymbol{p}$ die a priori Wahrscheinlichkeiten zusammenfasst, hat daher $\boldsymbol{S}$ die Form

$$\boldsymbol{S} = \left(\begin{array}{c:c|c} 1 & E\{\boldsymbol{\varphi}_{\text{rest}}(\boldsymbol{c})\}^{\mathsf{T}} & \boldsymbol{p}^{\mathsf{T}} \\ \hdashline E\{\boldsymbol{\varphi}_{\text{rest}}(\boldsymbol{c})\} & E\{\boldsymbol{\varphi}_{\text{rest}}(\boldsymbol{c})\,\boldsymbol{\varphi}_{\text{rest}}(\boldsymbol{c})^{\mathsf{T}}\} & E\{\boldsymbol{\varphi}_{\text{rest}}(\boldsymbol{c})\boldsymbol{\delta}(\boldsymbol{c})^{\mathsf{T}}\} \\ \hline \boldsymbol{p} & E\{\boldsymbol{\delta}(\boldsymbol{c})\boldsymbol{\varphi}_{\text{rest}}(\boldsymbol{c})^{\mathsf{T}}\} & \mathbf{diag}(p_\kappa) \\ \multicolumn{2}{c|}{\underbrace{\hphantom{xxxxxxxxxxxxxxxxxxxx}}_{m}} & \underbrace{\hphantom{xxxxxxxx}}_{k} \end{array} \right) \tag{4.4.43}$$

Mit der weiteren Matrix

$$\boldsymbol{T} = \begin{pmatrix} [E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathsf{T}}\}]^{-1} & \mathbf{0} \\ -E\{\boldsymbol{\delta}\boldsymbol{\varphi}^{\mathsf{T}}\}[E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathsf{T}}\}]^{-1} & \boldsymbol{I} \end{pmatrix} \tag{4.4.44}$$

und mit Berücksichtigung von (4.4.19) gilt

$$\begin{aligned} \boldsymbol{U} = \boldsymbol{T}\,\boldsymbol{S} &= \begin{pmatrix} \boldsymbol{I} & [E\{\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathsf{T}}\}]^{-1}E\{\boldsymbol{\varphi}\boldsymbol{\delta}^{\mathsf{T}}\} \\ \mathbf{0} & E\{\boldsymbol{\delta}\boldsymbol{\delta}^{\mathsf{T}}\} - E\{\boldsymbol{\delta}\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{A}\} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{I} & \boldsymbol{A}^* \\ \mathbf{0} & \boldsymbol{K}_{\boldsymbol{\epsilon}} \end{pmatrix} . \end{aligned} \tag{4.4.45}$$

Dabei ist $\boldsymbol{K}_{\boldsymbol{\epsilon}}$ die Kovarianzmatrix des Fehlervektors

$$\boldsymbol{\epsilon} = \boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi} , \tag{4.4.46}$$

die wegen (4.4.22) definiert ist mit

$$\begin{aligned} \boldsymbol{K}_{\boldsymbol{\epsilon}} &= E\left\{(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})^{\mathsf{T}}\right\} \\ &= E\left\{\boldsymbol{\delta}(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})^{\mathsf{T}} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi}(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})^{\mathsf{T}}\right\} \\ &= E\left\{\boldsymbol{\delta}(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})^{\mathsf{T}}\right\} - \boldsymbol{A}^{\mathsf{T}}E\left\{\boldsymbol{\varphi}(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})^{\mathsf{T}}\right\} \\ &= E\left\{(\boldsymbol{\delta}(\boldsymbol{\delta} - \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\varphi})^{\mathsf{T}}\right\} . \end{aligned} \tag{4.4.47}$$

Die letzte Zeile der obigen Gleichung folgt für die optimale Parametermatrix aus (4.4.21). Wenn man also die linken m Spalten der Matrix $\boldsymbol{S}$ normiert, entsteht in den rechten k Spalten die gesuchte optimale Parametermatrix $\boldsymbol{A}^*$. Zur Normierung sind, wie oben ausgeführt, die obigen Schritte 1 und 2 des GAUSS-JORDAN-Algorithmus hinreichend, nämlich Multiplikation einer Zeile mit einer Konstanten und Addition des Vielfachen einer Zeile zu einer anderen Zeile. Ein wesentlicher Vorteil der schrittweisen Umformung besteht darin, dass nach jeder Normierung einer weiteren Spalte (nach jedem „Schritt") eine Matrix $\boldsymbol{U}'$ entsteht von der in Bild 4.4.1 gezeigten Form. Dort ist zunächst der Einfachheit halber angenommen, dass die Spalten in ihrer üblichen Reihenfolge, d. h. ohne Pivotisierung, normiert werden.

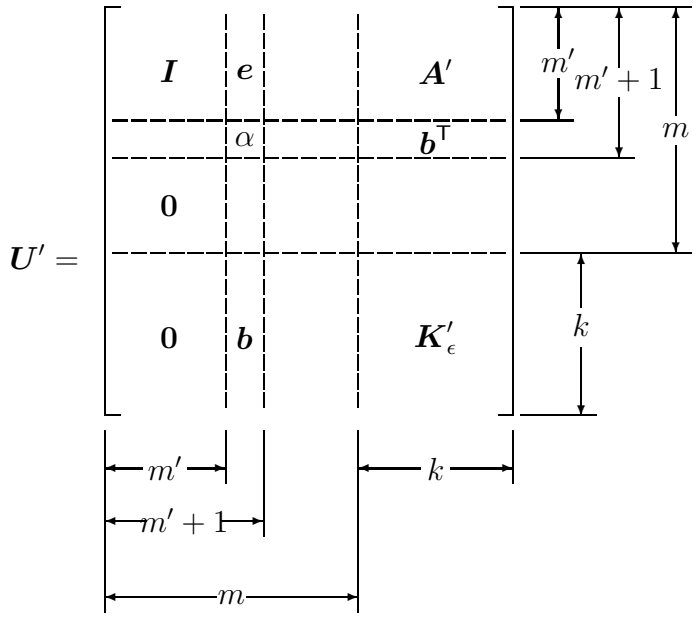


Bild 4.4.1: Zur schrittweisen Berechnung der Matrix $\boldsymbol{U}$ in (4.4.45)

Wenn von den m linken Spalten erst m' normiert sind, ist die $m' \times k$ Matrix $\boldsymbol{A}'$ ein Zwischenergebnis, um eine Trennfunktion $\boldsymbol{d}'$ mit Hilfe der m' Komponenten von $\boldsymbol{\varphi}$ zu berechnen, und $\boldsymbol{K}'_\epsilon$ ist die Kovarianzmatrix des dabei entstehenden Fehlervektors. Man kann also die Rechnung nach einer beliebigen Zahl von Schritten abbrechen und auf diese Weise statt der m Komponenten von $\boldsymbol{\varphi}$ nur $m' < m$ zur Berechnung der Trennfunktion verwenden. Die Darstellung in Bild 4.4.1 bedeutet natürlich nicht, dass die Normierung der Spalten in der durch die Komponenten von $\boldsymbol{\varphi}$ vorgegebenen Reihenfolge ausgeführt werden muss. Tatsächlich kann in jedem Schritt *irgendeine* der noch nicht normierten Spalten ausgewählt werden, wie es bei der Pivotisierung üblich ist. Dieses ist nochmals in der Gleichung

$$\boldsymbol{U}' = \left(\begin{array}{ccccc|c} \boldsymbol{I}^{m'\times m'} & \# & \boldsymbol{e}^{m'\times 1} & \# & & \boldsymbol{A}^{m'\times k} \\ \mathbf{0} & \# & \# & \# & & \# \\ \mathbf{0} & \# & \alpha & \# & & \boldsymbol{b}^{\mathsf{T}\,1\times k} \\ \mathbf{0} & \# & \# & \# & & \# \\ \hline \mathbf{0}^{k\times m'} & \# & \boldsymbol{b}^{k\times 1} & \# & & \boldsymbol{K}'^{\,k\times k}_\epsilon \\ \underbrace{\qquad\qquad\qquad\qquad}_{m} & & & & & \underbrace{\qquad}_{k} \end{array}\right)^{(m+k)\times(m+k)} \tag{4.4.48}$$

angedeutet, wobei α das aktuell ausgewählte Pivotelement ist und mit dem Symbol $\#$ Teilmatrizen bezeichnet sind, die in dem aktuellen Rechenschritt keine Rolle spielen.

In jedem Falle hat die Normierung einer weiteren $(m'+1)$-ten Spalte folgende Auswirkungen:

1. Die resultierende Matrix $\boldsymbol{U}''$ hat $(m'+1)$ normierte Spalten.
2. Die Matrix $\boldsymbol{A}'$ wächst um eine weitere Zeile und ergibt $\boldsymbol{A}''$.
3. Die Elemente der Matrix $\boldsymbol{K}'_\epsilon$ werden verändert in $\boldsymbol{K}''_\epsilon$.

Es kommt nun darauf an, den Rechenvorgang so steuern, dass lineare Abhängigkeiten beseitigt werden und die Abnahme des Fehlers ε in jedem Schritt möglichst groß ist.

Der Fehler ε in (4.4.14) ist das Optimierungskriterium für den Polynomklassifikator. Wenn man sukzessive die Spalten, und damit die Komponenten der Approximationsfunktion $\boldsymbol{\varphi}$ in (4.4.7), auswählt, die den Fehler am meisten verkleinern, so werden damit *systematisch* die besten Merkmale sowie deren Produktterme für den Polynomklassifikator ermittelt. Damit ist ein analytisches Verfahren zur Merkmalsauswahl für einen bestimmten Klassifikator realisiert; zwei weitere Beispiele für analytische Verfahren wurden in Abschnitt 3.8 vorgestellt.

Der Kürze halber geben wir nur die Gleichungen zur Berechnung der neuen Matrix $\boldsymbol{A}''$ und der veränderten Fehlermatrix $\boldsymbol{K}_\epsilon''$ sowie zur Auswertung der beiden Kriterien an; die Einzelheiten sind in der in Abschnitt 4.11 erwähnten Literatur enthalten. Es gilt mit den Bezeichnungen von Bild 4.4.1 oder (4.4.48)

$$\boldsymbol{A}'' = \begin{pmatrix} \boldsymbol{A}' \\ \boldsymbol{0}^\mathsf{T} \end{pmatrix} - \frac{1}{\alpha} \begin{pmatrix} \boldsymbol{e} \\ 1 \end{pmatrix} \boldsymbol{b}^\mathsf{T} , \tag{4.4.49}$$

$$\boldsymbol{K}_\epsilon'' = \boldsymbol{K}_\epsilon' - \frac{1}{\alpha} \boldsymbol{b}\boldsymbol{b}^\mathsf{T} . \tag{4.4.50}$$

Die Gleichungen (4.4.49) und (4.4.50) beschreiben die Änderungen der Ausgangsmatrizen $\boldsymbol{A}$, $\boldsymbol{K}_\epsilon$ durch die Rechenschritte. Insbesondere wird die für m' Terme gültige optimale Parametermatrix durch Hinzunahme einer weiteren Zeile wieder verändert. Die Normierung der Spalten von $\boldsymbol{S}$, bzw. einer weiteren $(m'+1)$-ten Spalte von $\boldsymbol{U}'$, wird mit den Schritten 1) und 2) des GAUSS-JORDAN-Algorithmus durchgeführt.

Zur Auswahl der nächsten zu normierenden Spalte wird zunächst die *lineare Abhängigkeit* verwendet. Eine weitere Komponente φ_μ ist dann nutzlos, wenn sie linear von den schon verwendeten $\varphi_1, \ldots, \varphi_{m'}$ abhängt, wie auch aus (3.9.41), S. 254, in Abschnitt 3.9.2 hervorgeht. Es lässt sich zeigen, dass das Hauptdiagonalelement α in Bild 4.4.1 bei linearer Abhängigkeit verschwindet. Als nächste zu normierende Spalte wird also die mit dem größten Wert von α genommen; dieses entspricht gerade der *Pivotisierung* durch Auswahl des größten Hauptdiagonalelements aus dem obigen Schritt 3 des GAUSS-JORDAN-Algorithmus bzw. der Vorgehensweise im Zusammenhang mit (4.4.40). Dort wird auf der Matrix $(\boldsymbol{B}\,|\,\boldsymbol{D})$ gearbeitet, hier auf der Matrix $\boldsymbol{S}$.

Ein anderes Auswahlkriterium ist die *Verminderung des Approximationsfehlers* ε, die sich durch Hinzunahme einer weiteren Komponente ergibt. Die Verminderung des Fehlers ist

$$\Delta\varepsilon = \frac{1}{\alpha} \boldsymbol{b}^\mathsf{T}\boldsymbol{b} . \tag{4.4.51}$$

Dieses folgt aus (4.4.50), da der Approximationsfehler ε in (4.4.15) gleich der Spur der Kovarianzmatrix in (4.4.50) ist. Als nächste zu normierende Spalte wird in diesem Falle also die mit dem größten Wert von $\Delta\varepsilon$ genommen. Das zweite Auswahlkriterium erfordert zwar zusätzlichen Rechenaufwand, ergibt aber die schnellste Abnahme des Approximationsfehlers ε. Zur Reduzierung linearer Abhängigkeiten kann zusätzlich zur Auswahl der nächsten Spalte, d. h. zusätzlich zur Pivotisierung mit (4.4.51), in jedem Rechenschritt noch überprüft werden, ob es zu kleine Hauptdiagonalelemente gibt; diese kennzeichnen Terme mit starker linearer Abhängigkeit zu schon ausgewählten Termen. Die zugehörigen Polynomterme werden durch Streichen der Zeilen und Spalten, in der die zu kleinen Hauptdiagonalelemente liegen, eliminiert. Das erfordert einen Schwellwert für die Mindestgröße eines Hauptdiagonalelements.

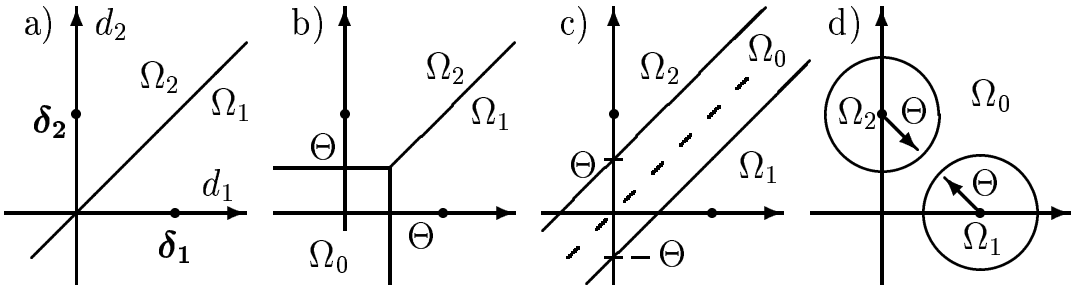


Bild 4.4.2: Verschiedene Rückweisungskriterien in der (d_1, d_2)-Ebene

Wegen der Wahl der Funktionen $\boldsymbol{\varphi}$ in (4.4.5) ist deren erste Komponente, wie erwähnt, $\varphi_1(\boldsymbol{c}) = 1$. Daher ist das links oben liegende Element der Matrix $\boldsymbol{S}$ in (4.4.42) bzw. (4.4.43) $s_{1,1} = 1$, und zwar unabhängig von den Daten der Stichprobe. Es ist daher zweckmässig, als erstes die erste Spalte zu normieren, erst in folgenden Rechenschritten die Pivotisierung nach maximaler Abnahme des Approximationsfehlers und ggf. die Beseitigung zu stark linear abhängiger Terme zu nutzen .

4.4.5 Rückweisungskriterium

Es gibt verschiedene heuristische Ansätze, ein Muster als nicht genügend zuverlässig klassifizierbar zurückzuweisen, die in Bild 4.4.2 für den Fall zweier Klassen veranschaulicht sind. Bild 4.4.2a zeigt die Aufteilung der (d_1, d_2)-Ebene für (4.4.1) ohne Rückweisung. Gemäß (4.4.11) werden k Trennfunktionen d_λ berechnet – im Bild ist $k = 2$ – und das Muster der Klasse mit maximalem d_λ zugeordnet. Die Aufteilung der (d_1, d_2)-Ebene in die Klassenbereiche Ω_1 und Ω_2 erfolgt durch die Gerade $d_1 = d_2$. Die ideale Trennfunktion für Muster aus Ω_1 ist nach (4.4.8), (4.4.11) durch die Konstante $\boldsymbol{\delta}_1 = (1, 0)^\mathsf{T}$ gegeben und für Muster aus Ω_2 durch $\boldsymbol{\delta}_2 = (0, 1)^\mathsf{T}$.

Ein erstes Rückweisungskriterium ergibt sich aus (4.1.25) und (4.4.18). Dividiert man nämlich (4.1.25), S. 312, durch $p(\boldsymbol{c})$, so folgt als optimale Rückweisungsregel

$$\text{wenn } \; p(\Omega_\kappa|\boldsymbol{c}) < \beta(\boldsymbol{c}) \; , \; \text{dann } \; \boldsymbol{c} \in \Omega_0 \; . \tag{4.4.52}$$

Analog wird (4.4.1) verallgemeinert zu

$$\begin{array}{l} \text{ermittle Index } \kappa \text{ mit } d_\kappa(\boldsymbol{c}) = \max_\lambda d_\lambda(\boldsymbol{c}) \\ \text{wenn } \; d_\kappa > \Theta \; , \; \text{dann } \; \boldsymbol{c} \in \Omega_\kappa \; , \; \text{sonst } \; \boldsymbol{c} \in \Omega_0 \; . \end{array} \tag{4.4.53}$$

Die Aufteilung der (d_1, d_2)-Ebene zeigt Bild 4.4.2b.

Das zweite Rückweisungskriterium wird analog (4.2.131) eingeführt, indem man festlegt

$$\begin{array}{l} \text{ermittle Index } \kappa 1 \text{ mit } d_{\kappa 1}(\boldsymbol{c}) = \max_\lambda d_\lambda(\boldsymbol{c}) \\ \text{und Index } \kappa 2 \text{ mit } d_{\kappa 2}(\boldsymbol{c}) = \max_{\lambda \neq \kappa 1} d_\lambda(\boldsymbol{c}) \; ; \\ \text{wenn } \; d_{\kappa 1} - d_{\kappa 2} > \Theta \; , \; \text{dann } \; \boldsymbol{c} \in \Omega_{\kappa 1} \; , \; \text{sonst } \; \boldsymbol{c} \in \Omega_0 \; . \end{array} \tag{4.4.54}$$

Bild 4.4.2c zeigt die Aufteilung der Ebene.

Das letzte Rückweisungskriterium schließlich wird aus der Forderung (4.4.14) abgeleitet, die ideale Trennfunktion möglichst gut zu approximieren. Dieser Gesichtspunkt spielt bei (4.4.53), (4.4.54) keine Rolle. Wir bezeichnen mit $\boldsymbol{\delta}_\kappa$ einen Vektor, dessen κ-te Komponente Eins ist, während alle anderen Null sind. Die Klassifikation erfolgt nach der Vorschrift

$$\begin{aligned} &\text{ermittle den Index } \kappa \text{ mit } d_\kappa(\boldsymbol{c}) = \max_\lambda d_\lambda(\boldsymbol{c}) \ , \\ &\text{berechne } \varepsilon(\boldsymbol{c}) = (\boldsymbol{\delta}_\kappa - \boldsymbol{d}(\boldsymbol{c}))^2 \ , \\ &\text{wenn } \ \varepsilon(\boldsymbol{c}) < \Theta^2 \ , \ \text{dann } \ \boldsymbol{c} \in \Omega_\kappa \ , \ \text{sonst } \ \boldsymbol{c} \in \Omega_0 \ . \end{aligned} \tag{4.4.55}$$

Es muss also der Abstand zwischen der Trennfunktion $\boldsymbol{d}$ in (4.4.13) und der durch die Entscheidung (4.4.1) ausgewählten idealen Trennfunktion $\boldsymbol{\delta}_\kappa$ kleiner als ein Schwellwert Θ bleiben. Natürlich sind die in (4.1.8) – (4.4.55) einheitlich mit Θ bezeichneten Schwellwerte i. Allg. verschieden. Die Aufteilung der (d_1, d_2)-Ebene zeigt Bild 4.4.2d. Dieses Kriterium ist auch nach experimentellen Befunden den anderen vorzuziehen.

4.5 Neuronale Netze (VA.2.2.3, 13.04.2004)

Die (künstlichen) **neuronalen Netze** (NN) sind geeignet, praktisch beliebige Funktionen zu approximieren, und sind damit ausgezeichnete Kandidaten für die Realisierung von Klassifikatoren. Einige ihrer Eigenschaften sind in Anlehnung an die (natürlichen) neuronalen Netze im zentralen Nervensystem von Lebewesen, insbesondere von Wirbeltieren, festgelegt worden. Es wurden unterschiedliche Typen von (künstlichen) neuronalen Netzen entwickelt. Von diesen werden im Folgenden das Mehrschicht-Perzeptron , die Netze mit radialen Basisfunktionen und die Merkmalskarte (KOHONEN-Karte) vorgestellt.

4.5.1 Vorbemerkungen

Das (natürliche) Neuron wird als elementare informationsverarbeitende Einheit angesehen. Es liefert über sein *Axon* in der Regel eine bestimmte Grundaktivität in Form von elektrischen Impulsen. Diese werden über ein Netzwerk von Verzweigungen und *synaptischen Verbindungen* an andere Neurone weitergegeben. Wenn die Eingangserregung eines Neurons einen bestimmten Betrag übersteigt, führt dieses zu einer signifikanten Änderung seiner eigenen Aktivität – in der Regel eine erhöhte Impulsfrequenz. Die Neurone zusammen mit ihren Verbindungen bilden ein neuronales Netz. Auch aus technischer Sicht wichtige Eigenschaften solcher Netze sind:

- *Lernfähigkeit*, d. h. die selbständige Veränderung des Verhaltens zur optimalen Reaktion auf neue Sachverhalte;
- *Adaptivität*, d. h. die selbständige Veränderung des Verhaltens zur optimalen Anpassung an sich zeitlich verändernde Sachverhalte;
- *Parallelität*, d. h. die parallele und verteilte Verarbeitung von Information.

Mit diesen Eigenschaften ist es möglich, aus Sicht der Informationsverarbeitung, speziell der Mustererkennung, wichtige Fähigkeiten in Echtzeit zu realisieren. Dazu gehören das Sehen, d. h. die Interpretation von Bildern der dreidimensionalen, zeitlich veränderlichen Welt; das Hören, d. h. die Interpretation akustischer Eindrücke, insbesondere von gesprochener Sprache; die motorische Steuerung, d. h. insbesondere die zielgerichtete Bewegung in der dreidimensionalen Welt; die reaktive Kopplung von Sensorik und Aktorik, d. h. die unmittelbare motorische Reaktion auf Sinneseindrücke, insbesondere auf gesehenes.

In einem (künstlichen) neuronalen Netz wird versucht, wesentliche Eigenschaften des natürlichen Vorbilds so nachzubilden, dass die oben genannten Eigenschaften und Fähigkeiten möglichst erhalten bleiben. Es wurden sowohl „feuernde“ (d. h. eine Impulsfolge als Ausgabesignal liefernde) als auch „nichtfeuernde“ (d. h. eine Spannung bzw. einen Funktionswert als Ausgabesignal liefernde) Neuronenmodelle entwickelt. Hier werden nur Netze mit „nichtfeuernden“ Neuronen vorgestellt. Ein neuronales Netz ist dann definiert durch

- die Menge von *Verarbeitungseinheiten* (Knoten, Neuronen);
- den Typ der *Berechnung*, die eine Einheit ausführt; in der Regel führen alle Einheiten eines Netzes Berechnungen vom gleichen Typ aus;
- die Menge der Verbindungen, welche die *Netzwerkstruktur* oder *Netzwerktopologie* definieren.

Ein Beispiel für ein einfaches Modellneuron, das sog. Schwellwertelement, zeigt Bild 4.5.1. Es erhält Eingabewerte x_i, die mit Gewichten w_i multipliziert und aufsummiert werden. Das

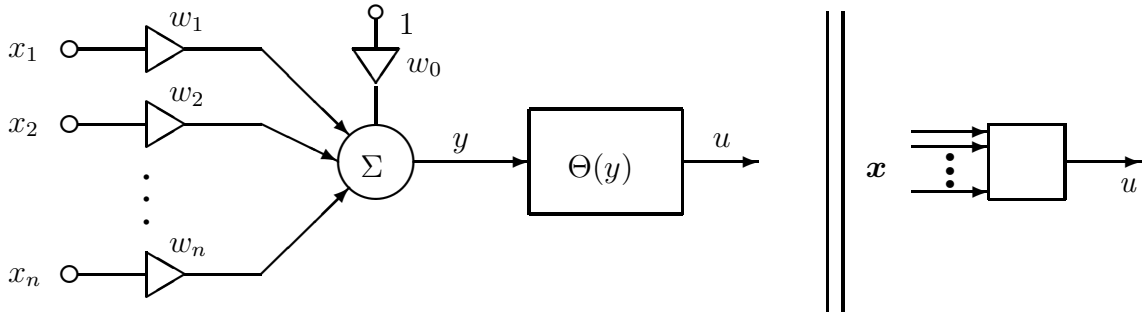


Bild 4.5.1: Ein Beispiel für ein Modellneuron, das Schwellwertelement

Ergebnis y wird durch eine nichtlineare Funktion Θ verarbeitet und ergibt die Ausgabe u. Da die nichtlineare Funktion in der Regel vom Typ eines Schwellwertes ist, wird dieses Modell auch als Schwellwertelement bezeichnet.

4.5.2 Mehrschicht-Perzeptron

Aufbau

Beim **Mehrschicht-Perzeptron** (abekürzt MLP für "multilayer perceptron") sind Knoten in Schichten geordnet. Verbindungen können nur von Knoten der einen Schicht zu Knoten der nächsten Schicht gehen, aber z. B. nicht zur vorhergehenden oder zur übernächsten. In der Regel sind die Schichten voll verbunden, d. h. von jedem Knoten der einen Schicht gehen Verbindungen zu jedem Knoten der nächsten Schicht.

Die *Eingabeschicht* erhält die Nummer $l = 0$ und hat $M^{(0)}$ Knoten für $M^{(0)}$ Eingabewerte $f_i^{(0)}$, $i = 0, 1, \ldots, M^{(0)} - 1$. Die Eingabewerte können z. B. Abtastwerte eines Musters oder auch die Werte von daraus extrahierten Merkmalen sein. In den Knoten der Eingabeschicht wird keine Verarbeitung durchgeführt. Es gibt eine Reihe von *verborgenen Schichten* ("hidden layers"), die jede $M^{(l)}$ Knoten haben und die weder eine Eingabe von ausserhalb des Netzes erhalten noch eine Ausgabe nach außen abgeben. Die Ausgabe des Knotens i in der Schicht l trägt zur Eingabe des Knotens j in der Schicht $(l + 1)$ mit dem Gewicht $w_{ij}^{(l+1)}$ bei. Die Gewichte sind zunächst unbekannt und werden durch *Training* des Netzes bestimmt. Die Eingaben in einen Knoten werden aufsummiert, über die Nichtlinearität geleitet und ergeben die Ausgabe eines Knotens. Das Endergebnis sind die Werte $f_j^{(L)}$, $j = 0, 1, \ldots, M^{(L)} - 1$ in der *Ausgabeschicht* mit der Nummer $l = L$ und mit $M^{(L)}$ Knoten. Bild 4.5.3 zeigt die Struktur und die verwendeten Bezeichnungen. Eingegebene Werte, z. B. Funktionswerte oder Komponenten eines Merkmalsvektors, werden durch die Schichten propagiert bis zur Ausgabe, die z. B. ein Klassenname ist, aber auch eine inverse Funktion sein kann.

Beispiele für Nichtlinearitäten sind die Funktionen

$$\Theta(y) = \frac{1}{1 + \exp[-\alpha y]} \qquad \text{(Sigmoid Funktion)}\,, \tag{4.5.1}$$

$$\Theta_{\mathrm{t}}(y) = \tanh[y] \qquad \text{(Tangens hyperbolicus)}\,, \tag{4.5.2}$$

$$= \frac{\exp[y] - \exp[-y]}{\exp[y] + \exp[-y]}\,,$$

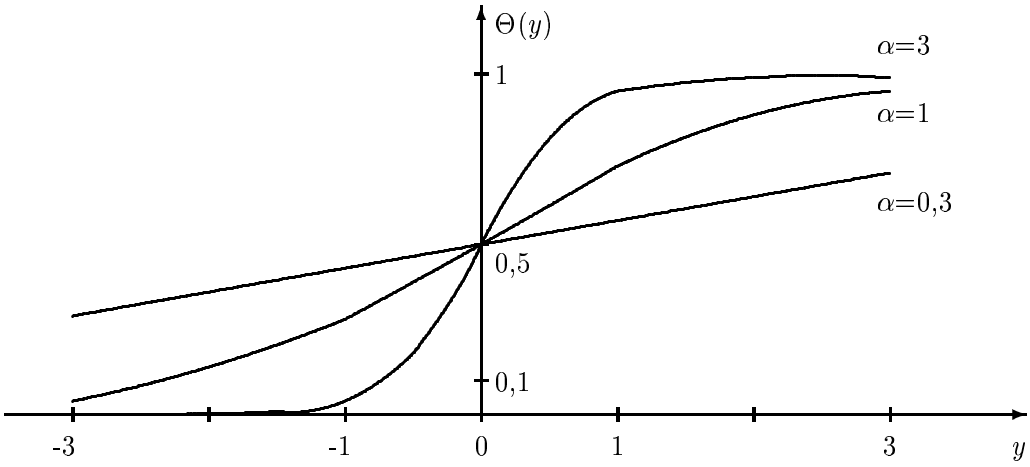


Bild 4.5.2: Die Sigmoid Funktion für drei Werte von α

$$\Theta_{\mathrm{s}}(y) \;=\; \begin{cases} 1 & : \quad y \geq 0 \\ 0 & : \quad y < 0 \end{cases} \qquad \text{(Schwellwertfunktion)} \; . \tag{4.5.3}$$

Die **Sigmoid Funktion** (4.5.1) ist in Bild 4.5.2 gezeigt. Sie ist, speziell auch mit $\alpha = 1$, eine übliche Wahl für die Nichtlinearität und lässt sich als Approximation eines Schwellwertes durch eine differenzierbare Funktion auffassen. Die Differenzierbarkeit ist für das unten gezeigte Training wichtig. Man erhält für die Ableitung

$$\frac{\mathrm{d}\Theta(y)}{\mathrm{d}y} = \alpha\Theta(y)\left(1-\Theta(y)\right) \; . \tag{4.5.4}$$

Vom Anwender eines MLP sind eine Reihe von Entwurfsentscheidungen zu treffen. Die Eingabegrößen bzw. die *Merkmale* sind zu wählen; die Netzwerkstruktur muss festgelegt werden, d. h. wieviele verborgene Schichten gibt es und wieviele Knoten gibt es in einer verborgenen Schicht; eine bestimmte *Nichtlinearität* und ein *Trainingsverfahren* sind zu wählen; das Ergebnis muss geeignet *kodiert* werden, d. h. die Zahl der Ausgabeknoten und die Darstellung der Ausgabegröße muss gewählt werden. Die Klärung dieser Fragen erfolgt in der Regel durch vergleichende Experimente. Das Problem der Festlegung geeigneter Merkmale ist, wie in Kapitel 3 erläutert, *das* Standardproblem der Mustererkennung. Zum Training der unbekannten Gewichte gibt es eine Reihe von Algorithmen, insbesondere die Fehlerrückführung in Satz 4.17. Für die Definition, Berechnung und das Training von MLP und anderen neuronalen Netzen gibt es verschiedene Softwarewerkzeuge, von denen einige auch Optionen zur automatischen Optimierung der Netzwerkstruktur enthalten. Die verfügbaren Softwarewerkzeuge, die sehr guten Leistungen bei der Klassifikation von Mustern (nach sorgfältiger experimenteller Festlegung obiger Fragen) sowie die in Satz 4.18 zusammengefassten prinzipiellen Eigenschaften von MLP haben diese zu einem wichtigen Ansatz zur Klassifikation von Mustern gemacht.

Berechnung

Die Berechnungen des MLP erfolgen schichtweise von der Eingabe– zur Ausgabeschicht. Man berechnet je Knoten (Neuron) in der Schicht $l+1$ die Gesamterregung $y_j^{(l+1)}$ als *gewichtete Summe* der Eingaben von der vorherigen Schicht. Das Ergebnis wird über eine *Nichtlinearität* Θ

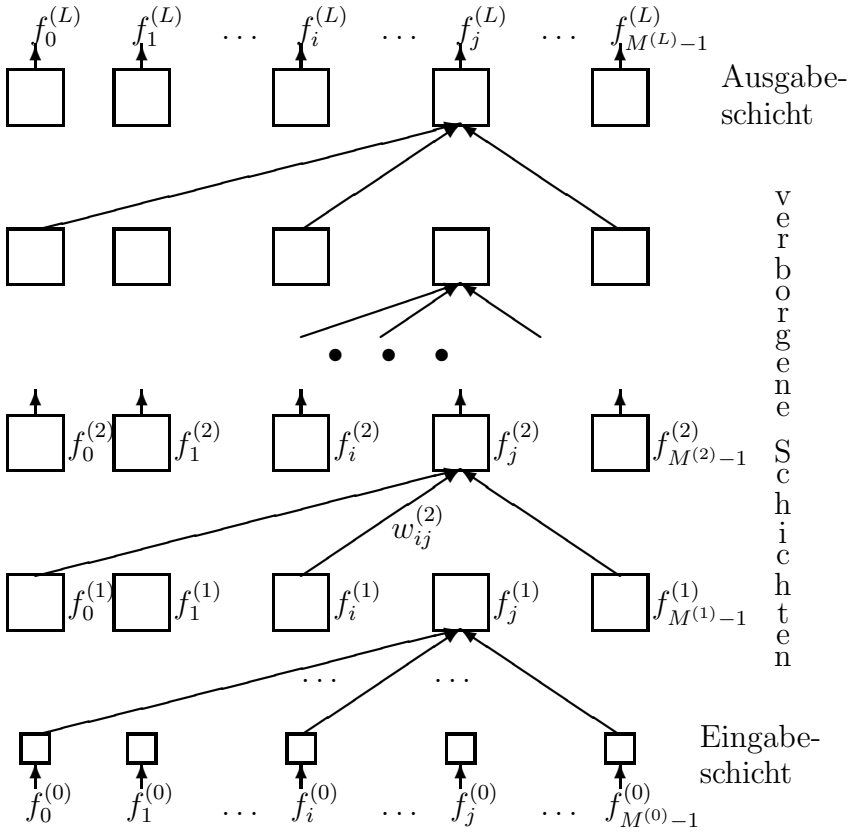


Bild 4.5.3: Die Struktur eines Mehrschicht-Perzeptron

geleitet, um die *Ausgabe* $f_j^{(l+1)}$ des Knotens zu bestimmen. Als Nichtlinearität Θ wird z. B. die *Sigmoid–Funktion* (4.5.1) gewählt. Diese Rechnung wird wiederholt bis man die Ausgabewerte bzw. die Ausgabeschicht erreicht hat. Zusammengefasst ergeben sich die Gleichungen

$$\boxed{\begin{aligned} y_j^{(l+1)} &= \sum_{i=0}^{M^{(l)}-1} w_{ij}^{(l+1)} f_i^{(l)} - w_j^{(l+1)} \,, \\ f_j^{(l+1)} &= \Theta\left(y_j^{(l+1)}\right) \,, \quad 0 \le j \le M^{(l+1)} - 1 \,, \quad l = 0, 1, \ldots, L-1 \,. \end{aligned}} \tag{4.5.5}$$

Eine kompaktere Schreibweise der obigen Gleichungen ergibt sich durch die Zusammenfassung der Gewichte in Matrizen und der Ausgabewerte in Vektoren. Dazu werden die Gewichte zur Schicht $(l+1)$ in einer Matrix

$$\begin{aligned} \boldsymbol{W}^{(l+1)} &= \left(w_{i,j}^{(l+1)}\right)^{\mathsf{T}} \\ &= \begin{pmatrix} w_{0,0}^{(l+1)} & w_{1,0}^{(l+1)} & \cdots & w_{M^{(l)}-1,0}^{(l+1)} \\ w_{0,1}^{(l+1)} & w_{1,1}^{(l+1)} & \cdots & w_{M^{(l)}-1,1}^{(l+1)} \\ & & \vdots & \\ w_{0,M^{(l+1)}-1}^{(l+1)} & w_{1,M^{(l+1)}-1}^{(l+1)} & \cdots & w_{M^{(l)}-1,M^{(l+1)}-1}^{(l+1)} \end{pmatrix} \end{aligned} \tag{4.5.6}$$

angeordnet und die Einzelgewichte im Vektor

$$\boldsymbol{w}^{(l+1)} = \left(w_0^{(l+1)},\, w_1^{(l+1)},\, \ldots\, w_{M^{(l+1)}-1}^{(l+1)}\right)^{\mathsf{T}} \,. \tag{4.5.7}$$

Die Ausgaben in der Schicht $l+1$ ergeben den Vektor

$$\boldsymbol{f}^{(l+1)} = \left(f_0^{(l+1)},\, f_1^{(l+1)},\, \ldots\, f_{M^{(l+1)}-1}^{(l+1)}\right)^{\mathsf{T}} \,. \tag{4.5.8}$$

Damit erhält man als äquivalente Schreibweise für (4.5.5)

$$\boxed{\boldsymbol{f}^{(l+1)} = \Theta\left(\boldsymbol{W}^{(l+1)}\boldsymbol{f}^{l} - \boldsymbol{w}^{(l+1)}\right) \,, \quad l = 0, 1, \ldots, L-1 \,.} \tag{4.5.9}$$

Man kann weiter den Vektor $\boldsymbol{w}^{(l+1)}$ in die Definitionen einbeziehen und vereinbaren

$$\begin{aligned} \widetilde{\boldsymbol{f}} &= (\boldsymbol{f}^{\mathsf{T}}, -1)^{\mathsf{T}} \,, \\ \widetilde{\boldsymbol{W}} &= (\boldsymbol{W}\,|\,\boldsymbol{w}) \,, \\ \boldsymbol{f}^{(l+1)} &= \Theta\left(\widetilde{\boldsymbol{W}}^{(l+1)}\widetilde{\boldsymbol{f}}^{l}\right) \,, \quad l = 0, 1, \ldots, L-1 \,. \end{aligned} \tag{4.5.10}$$

Wie bei anderen numerischen Klassifikatoren ist also auch hier die schnelle (u. U. auch die parallele) Berechnung von Skalarprodukten bzw. von Vektoroperationen wichtig.

Training

Die zunächst unbekannten Gewichte $w_{ij}^{(l)}$ werden durch einen Trainings– oder Lernprozess aus einer Stichprobe von Mustern bestimmt. Das Prinzip ist die iterative Veränderung der Gewichte (s. (1.6.7), S. 36), sodass der Fehler zwischen der tatsächlichen und der vorgegebenen gewünschten Ausgabe minimiert wird. Es liegt also ein *überwachtes* Lernen vor. Die iterative Minimierung des Fehlers erfolgt in der Regel durch einen Gradientenabstieg. Wenn Muster klassifiziert werden sollen, so ist die gewünschte Ausgabe die Nummer der richtigen Klasse. Im Prinzip kann diese in der Ausgabeschicht beliebig kodiert werden, jedoch wird oft ein „1–aus–k“ Kode verwendet, d. h. es gibt $M^{(L)} = k$ Ausgabeknoten. Wenn die Klasse Ω_κ erkannt wird, hat der Knoten κ idealerweise den Ausgabewert Eins, alle anderen Null. In diesem Falle ist die gewünschte Ausgabe dann wie schon in (4.4.8) definiert

$$\text{wenn } \boldsymbol{c} \in \Omega_\kappa \,,\ \text{dann } \delta_\kappa(\boldsymbol{c}) = 1 \,,\ \text{sonst } \delta_\lambda(\boldsymbol{c}) = 0 \,,\ \text{für } \lambda \neq \kappa \,. \tag{4.5.11}$$

Eine Alternative, die das Training beschleunigen kann, ist die „weiche“ Definition der gewünschten Ausgabe mit

$$\text{wenn } \boldsymbol{c} \in \Omega_\kappa \,,\ \text{dann } \delta_\kappa(\boldsymbol{c}) = 0{,}9 \,,\ \text{sonst } \delta_\lambda(\boldsymbol{c}) = 0{,}1 \,,\ \text{für } \lambda \neq \kappa \,. \tag{4.5.12}$$

Beispiele für Fehlermaße sind

$$\varepsilon_{\mathrm{MSE}} = 0{,}5 \sum_{i=0}^{M^{(L)}-1} \left(\delta_i - f_i^{(L)}\right)^2 \,, \tag{4.5.13}$$

$$\varepsilon_{\mathrm{McC}} = -0{,}5 \sum_{i=0}^{M^{(L)}-1} \ln\left[1 - \left(\delta_i - f_i^{(L)}\right)\right]^2 \,, \tag{4.5.14}$$

$$\varepsilon_{\mathrm{CE}} = -\sum_{i=0}^{M^{(L)}-1} \left(\delta_i \log[f_i^{(L)}] + (1-\delta_i)\log[1-f_i^{(L)}]\right) , \tag{4.5.15}$$

$$\varepsilon_{\mathrm{CFM}} = \frac{1}{M^{(L)}-1} \sum_{\substack{\lambda=1 \\ \lambda\neq\kappa}}^{M^{(L)}} \frac{a}{1+\exp\left[-(b(f_\kappa^{(L)} - f_\lambda^{(L)}) + c)\right]} . \tag{4.5.16}$$

Der mittlere quadratische Fehler $\varepsilon_{\mathrm{MSE}}$ ist ein häufiges Fehlermaß, das auch schon in (4.4.9) verwendet wurde. Der MCCLELLAND-Fehler $\varepsilon_{\mathrm{McC}}$ hat Vorteile bei vielen Klassen, d. h. bei vielen Ausgabeknoten. Die Verwendung der Kreuzentropie $\varepsilon_{\mathrm{CE}}$ soll die Unterschiede zwischen tatsächlicher und idealer Verteilungsdichte der Ausgabewerte minimieren. Das Maß $\varepsilon_{\mathrm{CFM}}$ soll den Unterschied zwischen dem Ausgabewert für die richtige Klasse und den Ausgabewerten der anderen Klassen maximieren. Dabei ist $f_\kappa^{(L)}$ der Wert für den Ausgabeknoten, der die richtige Klasse kodiert, $f_\lambda^{(L)}$ der Wert der anderen, und a, b, c sind Parameter. Alle diese Maße hängen letztlich von der Topologie und den Gewichten des neuronalen Netzes ab.

Die Einstellung der Gewichte erfolgt durch einen Gradientenabstieg gemäß

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \beta\frac{\partial\varepsilon}{\partial w_{ij}^{(l)}} = w_{ij}^{(l)} + \Delta w_{ij}^{(l)} . \tag{4.5.17}$$

Dabei ist β die *Schrittweite* und $\partial\varepsilon/\partial w_{ij}^{(l)}$ die *Richtung* des Abstiegs. Die Schrittweite wird empirisch festgelegt. Der Gradientenabstieg ist für alle obigen Fehlermaße möglich.

Als Beispiel wird für den mittleren quadratischen Fehler $\varepsilon_{\mathrm{MSE}}$ in (4.5.13) und die Sigmoid Funktion in (4.5.1) mit $\alpha = 1$ die Differentiation in der Ausgabeschicht $l = L = 3$ durchgeführt. Man erhält mit der Kettenregel der Differentiation

$$\begin{aligned}
\frac{\partial\varepsilon_{\mathrm{MSE}}}{\partial w_{ij}^{(L)}} &= \frac{\partial\varepsilon_{\mathrm{MSE}}}{\partial f_j^{(L)}} \frac{\partial f_j^{(L)}}{\partial y_j^{(L)}} \frac{\partial y_j^{(L)}}{\partial w_{ji}^{(L)}} \\
&= \frac{\partial}{\partial f_j^{(3)}}\left(0,5\sum_i(\delta_i - f_i^{(3)})^2\right) \frac{\partial}{\partial y_j^{(3)}}\left(\frac{1}{1+\exp(-y_j^{(3)})}\right) \\
&\quad \frac{\partial}{\partial w_{ij}^{(3)}}\left(\sum_i w_{ij}^{(3)} f_i^{(2)} - w_j^{(3)}\right) \\
&= -\left(\delta_j - f_j^{(3)}\right)\left(1 - f_j^{(3)}\right) f_j^{(3)} f_i^{(2)} \\
&= -d_j^{(3)} f_i^{(2)}
\end{aligned} \tag{4.5.18}$$

Damit wird der Korrekturterm

$$\Delta w_{ij}^{(3)} = \beta d_j^{(3)} f_i^{(2)} . \tag{4.5.19}$$

Eine entsprechende Rechnung ergibt die Korrekturterme (4.5.22) für die unteren Schichten. Die Gleichungen für das Training sind nachfolgend zusammengefasst.

Satz 4.17 **(Fehlerrückführungs–Algorithmus)** *("error–back–propagation") Das Training des MLP erfolgt nach den Gleichungen:*

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} + \beta d_j^{(l)} f_i^{(l-1)} \quad l = L, L-1, \ldots, 1 \,, \tag{4.5.20}$$

$$d_j^{(L)} = \left(\delta_j - f_j^{(L)}\right)\left(1 - f_j^{(L)}\right) f_j^{(L)} \,, \tag{4.5.21}$$

$$d_j^{(l-1)} = \sum_{k=0}^{M^{(l)}-1} d_k^{(l)} w_{kj}^{(l)} \left(1 - f_j^{(l-1)}\right) f_j^{(l-1)} \quad l = L, L-1, \ldots, 2 \,. \tag{4.5.22}$$

Das Training wird iterativ durchgeführt, wobei sukzessive Muster $\boldsymbol{f}$ einer Stichprobe an der Eingabeschicht angeboten werden. Mit (4.5.5) bzw. (4.5.9) wird die Ausgabe des Netzes berechnet. Für die Klassifikation von Mustern wird vorausgesetzt, dass die Trainingsstichprobe klassifiziert ist, d. h. der Wert von $\delta_\kappa(\boldsymbol{c})$ in (4.5.11) oder (4.5.12) ist bekannt. Beginnend mit der Ausgabeschicht kann nun (4.5.21) ausgewertet werden und neue Gewichte $w_{ij}^{(L)}$ mit (4.5.20) berechnet werden. Dann werden schrittweise neue Gewichte in den unteren Schichten mit (4.5.22) berechnet. Der Fehler $(\delta_j - f_j^{(L)})$ wird also von der Ausgabeschicht zur Eingabeschicht „zurückgeführt", und daher kommt der Name für diesen Trainingsalgorithmus, nämlich Fehlerrückführungs–Algorithmus (bzw. "error–back–propagation algorithm").

Das Training eines neuronalen Netzes kann sehr langwierig sein. Daher wurden verschiedene Ansätze zur Beschleunigung des Trainings vorgeschlagen. Mit der Einführung eines **Momententerms** wird die Änderung der Gewichte modifiziert zu

$$\Delta w_{ij,N}^{(l)} = \beta d_{j,N}^{(l)} f_{i,N}^{(l-1)} + \gamma \Delta w_{ij,N-1}^{(l)} \,, \tag{4.5.23}$$

wobei β und γ Parameter sind, die experimentell festgelegt werden. Weitere Varianten des Trainings sind der zitierten Literatur zu entnehmen.

Eigenschaften

Da sich mit dem MLP beliebige Funktionen approximieren lassen, eignet es sich sowohl für die Klassifikation von Mustern als auch z. B. für die Vorhersage, Inversion oder Glättung von Funktionen. Einige wichtige Aussagen sind im Folgenden zusammengefasst, wobei für Beweise auf die angegebene Literatur verwiesen wird.

Satz 4.18 *Das MLP erlaubt*

1. *die Definition jeder logischen Funktion (zwei Schichten von Gewichten sind hinreichend,* Beweis*: s. z. B. [Muroga, 1971]);*
2. *die Approximation jeder nichtlinearen Funktion (zwei Schichten von Gewichten sind hinreichend,* Beweis*: s. z. B. [Hornik et al., 1989, White, 1990]);*
3. *die Definition* beliebiger Klassengrenzen *im $\mathbb{R}^n$ (zwei Schichten von Gewichten sind hinreichend,* Beweis*: s. z. B. [Makhoul et al., 1989]).*

Die Tatsache, dass zwei Schichten für die Approximationen hinreichend sind, schließt nicht aus, dass u. U. drei oder noch mehr Schichten eine schnellere Konvergenz des Trainings erlauben und daher vorzuziehen sind. Insbesondere für die Klassifikation werden oft $L = 3$ Schichten von Gewichten vorgeschlagen, wobei die Argumentation anschaulich darin besteht, dass

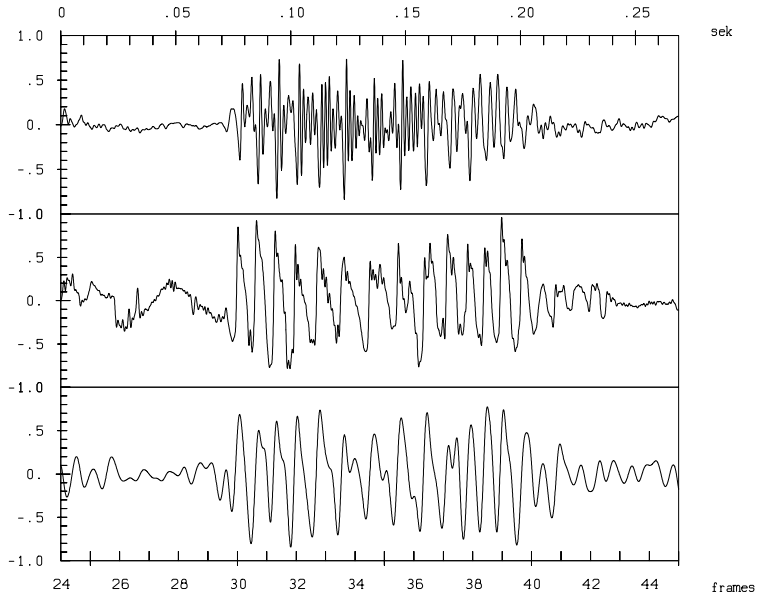


Bild 4.5.4: Das Bild zeigt oben ein Sprachsignal, in der Mitte das Laryngograph Anregungssignal, unten das mit einem MLP rekonstruierte Anregungssignal (aus [Denzler, 1992])

man mit der ersten Schicht eine Menge von Hyperebenen realisiert und mit der Nichtlinearität festlegt, auf welcher Seite der Hyperebene sich ein Muster befindet. In der zweiten Schicht werden Hyperebenen durch logisches UND zu einem konvexen Gebiet zusammengefasst, und in der dritten (bei Bedarf) mehrere konvexe Gebiete durch logisches ODER zu einem mehrfach zusammenhängenden Klassengebiet. Da die Zahl der Hyperebenen beliebig ist, kann man so beliebig viele mehrfach zusammenhängende Klassengebiete definieren. Für die Verwendung des MLP zur Approximation von Funktionen spricht, dass die Sigmoid Funktion – im Unterschied z. B. zu Polynomen – für betragsmäßig große Werte des Arguments *nicht* gegen Unendlich geht.

Ein Beispiel für eine „Funktionsinversion“ zeigt Bild 4.5.4. Es handelt sich dabei um die Schätzung der sprecherunabhängigen inversen Abbildung eines Sprachsignals auf das Anregungssignal der Stimmbänder bei stimmhafter Sprache. Dieses ist nützlich für die Bestimmung der Sprachgrundfrequenz. Hier ist die Eingabe des neuronalen Netzes ein Sprachsignal und die gewünschte Ausgabe das mit einem Laryngographen gemessene Anregungssignal, das durch die tatsächlich beobachtete Ausgabe approximiert wird. Ein Netz wurde mit verschiedenen Sprachsignalen trainiert und dann mit Sprachsignalen getestet, die *nicht* in der Trainingsmenge enthalten waren. Man entnimmt der Abbildung, dass das Anregungssignal sehr gut durch das neuronale Netz approximiert wird.

Von einem MLP mit zwei verborgenen Schichten lässt sich zeigen, dass es eine Stichprobe $\omega = \{(\boldsymbol{c}, \boldsymbol{t})\}$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{t} \in \mathbb{R}^m$ vom Umfang N mit sehr kleinem Fehler lernen kann, wenn das Netz in der ersten verborgenen Schicht $L_1 = \sqrt{(m+2)N} + 2\sqrt{N/(m+2)}$, in der zweiten $L_2 = m\sqrt{N/(m+2)}$ Neuronen hat sowie m Ausgabeneuronen hat. Für Einzelheiten wird auf die Literatur verwiesen.

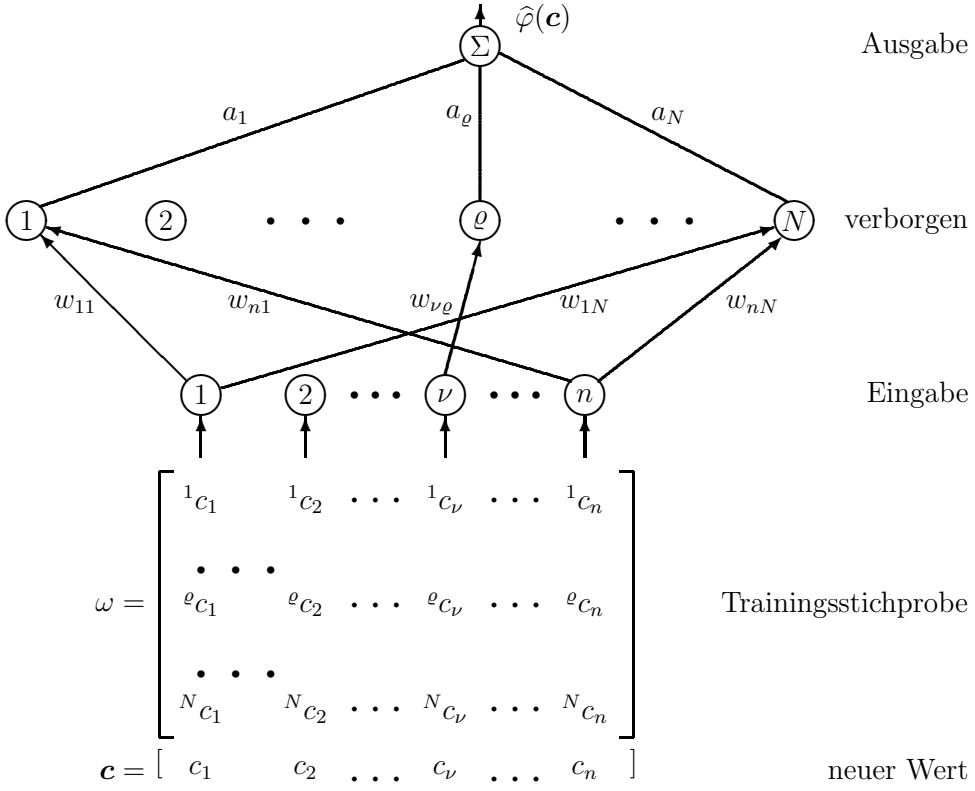


Bild 4.5.5: Ein Netz mit radialen Basisfunktionen. Unten ist die Trainingsstichprobe angedeutet; für die Gewichte gilt $w_{\nu\varrho} = {}^{\varrho}c_\nu$

4.5.3 Netze mit radialen Basisfunktionen

Bei diesem Netzwerktyp gibt es nur *eine* verborgene Schicht. Die Neuronen haben eine *radialsymmetrische* Aktivierungsfunktion, die als Basisfunktionen für die Approximation von Funktionen anhand von Stützstellen bzw. von Stichprobenwerten anzusehen sind. Diese **radialen Basisfunktionen** haben nur in der Nähe der Stützstellen merklich von Null verschiedene Werte. Damit wird erreicht, dass eine Stützstelle sich nur lokal und mit endlichem Wert auswirkt, während die Sigmoid Funktion (4.5.1) sich global und mit endlichem Wert auswirkt und ein Term einer Potenzreihe global und mit unbegrenzt wachsendem Wert. Die Netzstruktur zeigt Bild 4.5.5. Die Gewichte von der Eingabeschicht zur verborgenen Schicht werden so gewählt, dass am ϱ–ten Knoten der verborgenen Schicht die Komponenten der ϱ–ten Stützstelle ${}^{\varrho}\boldsymbol{c}$ (bzw. des ϱ–ten beobachteten Musters der Trainingsstichprobe ω) anliegen. Diese werden im verborgenen Knoten mit dem zu approximierenden Muster geeignet verrechnet. Wegen der einfachen Struktur der Netze ist damit die direkte Berechnung der noch verbleibenden Parameter a_ϱ möglich.

Im einfachsten Fall wird eine Funktion $\varphi(\boldsymbol{c})$, von der nur Werte ${}^{\varrho}\varphi = \varphi({}^{\varrho}\boldsymbol{c})$ an diskreten Stützstellen ${}^{\varrho}\boldsymbol{c}$, $\varrho = 1, \ldots, N$ bekannt sind, durch eine Funktion $\widehat{\varphi}(\boldsymbol{c})$ approximiert. Die Approximation erfolgt durch eine gewichtete Summe von radialen Basisfunktionen

$$g_\varrho(\boldsymbol{c}) \;=\; g\left(\|\boldsymbol{c} - {}^{\varrho}\boldsymbol{c}\|\right) \;, \tag{4.5.24}$$

wobei $\|\cdot\|$ z. B. der EUKLID-Abstand ist. Die Approximation ist

$$\widehat{\varphi}(\boldsymbol{c}) = \sum_{\varrho=1}^{N} a_\varrho \, g\left(||\boldsymbol{c} - {}^{\varrho}\boldsymbol{c}||\right) \ . \tag{4.5.25}$$

Diese Gleichung erinnert im Prinzip an die PARZEN-Schätzung in (4.2.142), S. 354. Eine mögliche Basisfunktion ist die Normalverteilung. Es wird gefordert, dass für eine Stützstelle (bzw. ein Trainingsmuster) ${}^{j}\boldsymbol{c}$ der Funktionswert ${}^{j}\varphi$ und die Approximation $\widehat{\varphi}({}^{j}\boldsymbol{c})$ übereinstimmen. Das liefert das lineare Gleichungssystem

$$\begin{aligned} {}^{j}\varphi &= \widehat{\varphi}({}^{j}\boldsymbol{c}) && (4.5.26)\\ &= \sum_{\varrho=1}^{N} a_\varrho \, g\left(||{}^{j}\boldsymbol{c} - {}^{\varrho}\boldsymbol{c}||\right) \ , \quad j = 1, \ldots, N && (4.5.27) \end{aligned}$$

für die unbekannten Koeffizienten a_ϱ. Die Terme $g\left(||{}^{j}\boldsymbol{c} - {}^{\varrho}\boldsymbol{c}||\right)$ werden im ϱ–ten Knoten berechnet. Um das zu ermöglichen werden die Gewichte festgelegt zu

$$w_{\nu\varrho} = {}^{\varrho}c_\nu \ . \tag{4.5.28}$$

Die noch verbleibenden Parameter a_ϱ ergeben sich mit (4.5.26) zu

$$\begin{pmatrix} a_1 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} g\left(||{}^{1}\boldsymbol{c} - {}^{1}\boldsymbol{c}||\right) & \ldots & g\left(||{}^{1}\boldsymbol{c} - {}^{N}\boldsymbol{c}||\right) \\ & \vdots & \\ g\left(||{}^{N}\boldsymbol{c} - {}^{1}\boldsymbol{c}||\right) & \ldots & g\left(||{}^{N}\boldsymbol{c} - {}^{N}\boldsymbol{c}||\right) \end{pmatrix}^{-1} \begin{pmatrix} {}^{1}\varphi \\ \vdots \\ {}^{N}\varphi \end{pmatrix} \ . \tag{4.5.29}$$

Mit (4.5.28) und (4.5.29) sind alle Parameter des Netzwerks bestimmt.

4.5.4 Merkmalskarte

Der als KOHONEN-Abbildung, **Merkmalskarte** oder auch *topologische Karte* bezeichnete Ansatz erlaubt ein *unüberwachtes Lernen* von Ähnlichkeiten in Beobachtungen, d. h. es ist *keine* klassifizierte Stichprobe zum Training erforderlich. Die Merkmalskarte stellt ähnliche Beobachtungsvektoren in *benachbarten Gebieten* dar, sodass Ähnlichkeiten auch visuell erkennbar sind.

Das Prinzip ist in Bild 4.5.6 gezeigt. Eine Menge von Ausgabegrößen (oder Ausgabeneuronen) $y_0, y_1, \ldots, y_{M'-1}$ wird in einem zweidimensionalen Gitter angeordnet. Jedem Ausgabeneuron y_j ist ein Gewichtsvektor $\boldsymbol{w}_j$ zugeordnet. Eine Beobachtung $\boldsymbol{f}$ (oder ein Merkmalsvektor $\boldsymbol{c}$) wird mit allen Gewichtsvektoren verglichen und mit der Minimumabstandsklassifikation wird der ähnlichste Gewichtsvektor $\boldsymbol{w}_\kappa$ ausgewählt, dessen zugehöriges Ausgabeneuron y_κ aktiviert wird. Durch ein geeignetes Training wird erreicht, dass Beobachtungen aus ähnlichen Klassen auf benachbarte Ausgaben abgebildet werden. Die Klassen werden nicht vom Entwickler vorgegeben, sondern im Training *unüberwacht* gelernt.

Wenn die Gewichtsvektoren trainiert sind, wird mit diesen die Ausgabe

$$y_j = \sum_{i=0}^{M-1} w_{ji} f_i = \boldsymbol{w}_j^{\mathsf{T}} \boldsymbol{f} \ , \quad 0 \le j \le M' - 1 \tag{4.5.30}$$

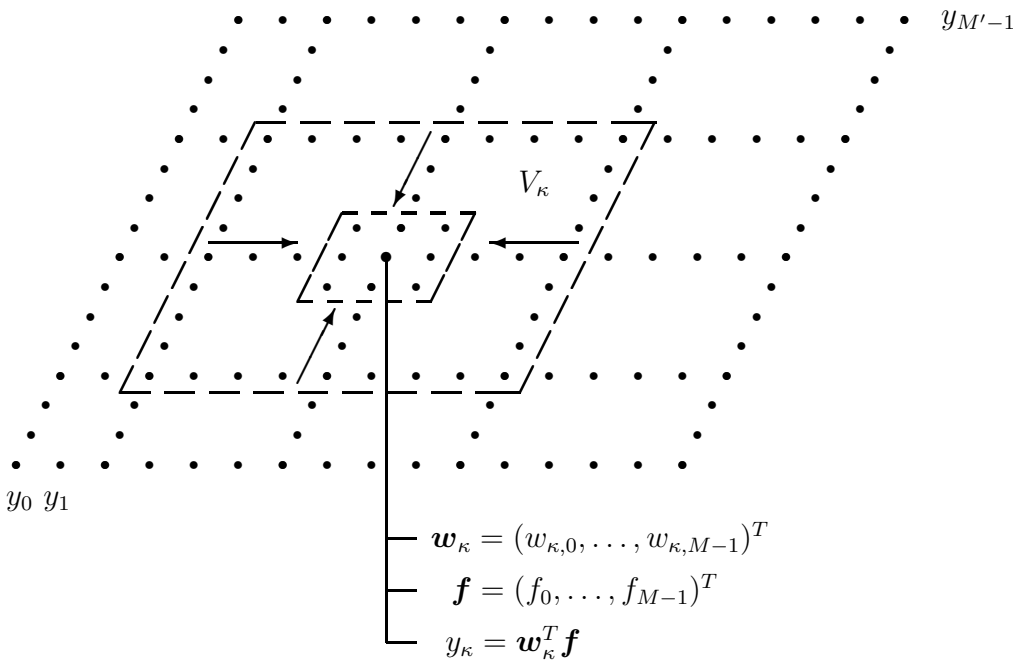


Bild 4.5.6: Eine Merkmalskarte (KOHONEN-Karte) mit Ausgabeneuronen $y_0, \ldots, y_{M'-1}$. Eine Beobachtung $\boldsymbol{f}$ wird mit jedem Gewichtsvektor $\boldsymbol{w}_j$ verglichen. Beim Training wird das Trainingsgebiet V_κ mit der Zeit eingeengt.

berechnet. Es wird das Ausgabeneuron y_κ bestimmt, dessen Gewichtsvektor $\boldsymbol{w}_\kappa$ am besten mit der Eingabe übereinstimmt, also

$$\boldsymbol{w}_\kappa = \underset{\boldsymbol{w}_j}{\operatorname{argmin}} |\boldsymbol{f} - \boldsymbol{w}_j|^2 \quad \Rightarrow \quad \text{aktiviere } y_\kappa \; . \tag{4.5.31}$$

Dieses ist als *Klassifikation* von $\boldsymbol{f}$ nach dem Minimumabstandsverfahren in eine Klasse Ω_κ aufzufassen, die durch den Gewichtsvektor $\boldsymbol{w}_\kappa$ definiert ist.

Zum **Training** werden die Gewichtsvektoren mit beliebigen Werten initialisiert, und es muss eine Trainingsstichprobe von Beobachtungen gegeben sein. Mit den aktuell gegebenen Gewichtsvektoren wird für eine Beobachtung $\boldsymbol{f}$ das aktivierte Neuron y_κ mit (4.5.31) bestimmt. Nun werden einige Gewichte durch die Kombination eines *Gradientenabstiegs* mit einer *Nachbarschaftsheuristik* so geändert, dass der Gewichtsvektor von y_κ der Beobachtung ähnlicher wird und dass Beobachtungen aus einer ähnlichen Klasse Ω_κ in die Nachbarschaft von y_κ abgebildet werden. Die Klassifikation der Beobachtungen und die Änderung der Gewichtsvektoren wird iteriert, bis die Gewichte sich nicht mehr (bzw. nicht mehr wesentlich) ändern. Um einen Gewichtsvektor dem Eingabevektor ähnlicher zu machen, wird der Fehler

$$\varepsilon = \sum_i (f_i - w_{ji})^2 \tag{4.5.32}$$

betrachtet. Der Gradientenabstieg zu seiner Minimierung liefert

$$w_{ji} \leftarrow w_{ji} - \beta \frac{\partial \varepsilon}{\partial w_{ji}} = w_{ji} + \beta(f_i - w_{ji}) \; . \tag{4.5.33}$$

Die Änderung der Gewichte nach dieser Gleichung erfolgt nicht nur für das Ausgabeneuron y_κ, sondern für alle Ausgabeneuronen y_j in einer Nachbarschaft V_κ. Zu Anfang des Trainings

wird eine relativ große Nachbarschaft gewählt, wie es in Bild 4.5.6 durch das große rechteckige Gebiet angedeutet ist. Alle Gewichtsvektoren $\boldsymbol{w}_j$ aus der Nachbarschaft V_κ werden mit (4.5.33) korrigiert. Im Laufe der Trainingsiterationen wird die Größe der Nachbarschaft reduziert, wie es das kleine Rechteck in Bild 4.5.6 andeutet. Das ergibt zusammengefasst die Trainingsvorschrift

1. wähle anfänglich eine relativ große Nachbarschaft V_κ ;
2. bestimme das aktivierte Neuron y_κ mit (4.5.31) ;
3. modifiziere die Gewichte in V_κ mit (4.5.33) ;
4. wiederhole die Schritte 2. und 3. mit abnehmender Nachbarschaft, bis die Gewichte sich nicht mehr (wesentlich) ändern .

(4.5.34)

man klassifiziere ein vorgelegtes Muster mit n_0 Merkmalen, wobei anfangs eine hohe Rückweisungsschwelle eingestellt wird		
	IF	das Muster wurde zurückgewiesen
	THEN	Klassifiziere mit $n_0 = n_0 + \Delta n$, $\Delta n \geq 1$ und ganzzahlig, Merkmalen, wobei die Rückweisungsschwelle erniedrigt wurde
UNTIL [das Muster ist in einer der Klassen eingeordnet] ODER [Die Zahl n_0 erreicht eine vorgegebene Größe $n_{\max}$]		

Bild 4.6.1: Prinzip der sequentiellen Klassifikation

4.6 Andere Klassifikatortypen (VA.1.1.3, 13.04.2004)

4.6.1 Sequentielle Klassifikatoren

Die in den vorigen Abschnitten beschriebenen Klassifikatoren verwenden stets eine feste Anzahl n von Merkmalen. Bei der Wahl von n muss ein Kompromiss zwischen Aufwand und Klassifikatorleistung geschlossen werden. Wenn man Muster zu klassifizieren hat, die nahe am Klassenzentrum (weit von der Klassengrenze) liegen, wird man vermutlich mit kleineren Werten von n auskommen als bei anderen. Ein **sequentieller Klassifikator** beginnt die Klassifikation mit wenigen Merkmalen, im Grenzfall $n = 1$, prüft dann, ob damit eine Klassifikation genügend zuverlässig möglich ist, und nimmt weitere Merkmale dazu, falls eine Klassifikation noch nicht möglich ist. Dieses Prinzip ist grundsätzlich bei statistischen, verteilungsfreien und nichtparametrischen Klassifikatoren anwendbar. Es ist zu erwarten, dass bei richtiger Anwendung *im Mittel weniger* Merkmale erforderlich sind. Allerdings erfordert sequentielle Klassifikation auch zusätzlich Maßnahmen, sodass im Einzelfall zu prüfen ist, ob dieses insgesamt lohnend ist.

Zu sequentiellen Methoden gibt es eine umfangreiche Literatur, jedoch ist die Bedeutung dieses Ansatzes für die Musterklassifikation gering geblieben. Die Vorgehensweise ist in Bild 4.6.1 skizziert. Sie lohnt sich insbesondere, wenn die Kosten für die Ermittlung eines Merkmals sehr hoch sind, wie z. B. in der medizinischen oder technischen Diagnostik.

Der sog. sequentielle Wahrscheinlichkeitstest ist ein Spezialfall, bei dem $n_0 = \triangle n = 1$, $k = 2$, $n_{\max} \rightarrow \infty$, feste Rückweisungsschranken und ein statistischer Klassifikator mit $p_1 = p_2$ verwendet werden. In diesem Spezialfall lässt sich zeigen, dass die sequentielle Vorgehensweise im Mittel die Zahl der zur Klassifikation erforderlichen Merkmale minimiert. Trotzdem bleibt das Problem, dass man die Folge der bedingten Dichten der $1-, 2-, ..., n_{\max}-$dimensionalen Merkmalsvektoren bestimmen und speichern muss. Für den Normalverteilungsklassifikator ist also die Folge der inversen Kovarianzmatrizen wachsender Größe zu speichern. Eine wesentliche Voraussetzung für die sinnvolle Anwendung sequentieller Verfahren ist offenbar, dass die mit n_0 Merkmalen berechneten Prüfgrößen – z. B. u_λ in (4.1.13), (4.2.122) oder d_λ in (4.4.6) – für eine iterative Berechnung der Prüfgrößen mit $n_0 + \triangle n$ Merkmalen herangezogen werden können. Das ist in den meisten Fällen möglich, trotzdem ist der Gesamtaufwand für die Realisierung eines Skalarproduktes wie in (4.2.122), (4.4.6) meistens geringer als der mit der sequentiellen Klassifikation verbundene. Bei den in Abschnitt 4.6.4 erörterten abstandsmessenden Klassifikatoren ist dagegen eine sequentielle Vorgehensweise ohne wesentlichen zusätzlichen Aufwand möglich. Der mit n Merkmalen berechnete Abstand $D^{(n)}$

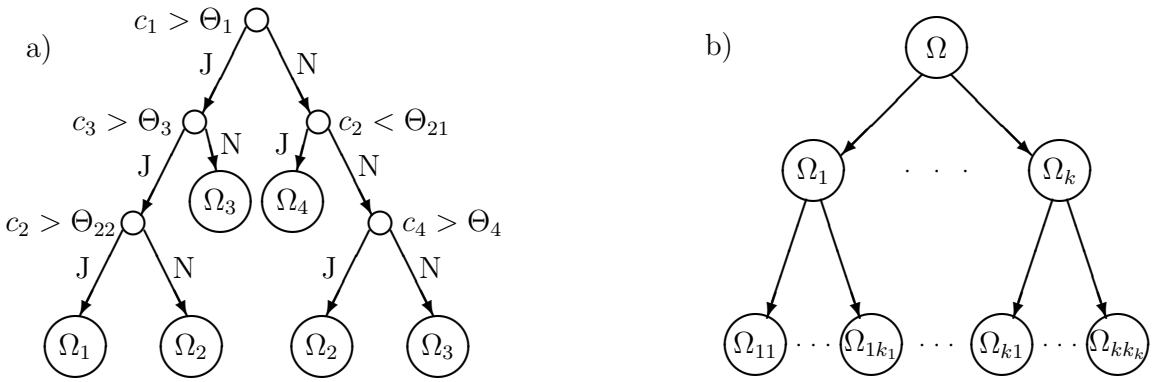


Bild 4.6.2: Zwei Beispiele für Klassifikationsbäume

zwischen einem neuen Merkmal $\boldsymbol{c}$ und einem Prototypen $\boldsymbol{c}_\lambda$ einer Klasse Ω_λ ist nämlich

$$\begin{aligned} D^{(n)} &= \sum_{\nu=1}^{n} (c_{\lambda\nu} - c_\nu)^2 \\ &= \sum_{\nu=1}^{n-1} (c_{\lambda\nu} - c_\nu)^2 + (c_{\lambda n} - c_n)^2 \\ &= D_\lambda^{(n-1)} + (c_{\lambda n} - c_n)^2 \,, \end{aligned} \tag{4.6.1}$$

d. h. der Abstand $D^{(n)}$ kann aus $D^{(n-1)}$ durch Addition eines weiteren Terms berechnet werden.

4.6.2 Klassifikationsbäume und hierarchische Klassifikation

Aus Bild 4.1.1, S. 310, und Bild 4.2.5, S. 350, geht hervor, dass die Klassifikation in einer Stufe durchgeführt wird. Aus Satz 4.1 bis Satz 4.3 geht hervor, dass diese Vorgehensweise auch in bestimmtem Sinne optimal ist. Trotzdem kann es praktisch zweckmäßig sein, eine Klassifikation in mehreren Stufen vorzunehmen. Dafür gibt es die in Bild 4.6.2 gezeigten, prinzipiell ähnlichen Ansätze. In Bild 4.6.2a wird ein binärer Klassifikationsbaum aufgebaut, bei dem in jedem Knoten ein Merkmal mit einem Schwellwert verglichen wird. Der **Klassifikationsbaum** (oder der *Entscheidungsbaum*) braucht i. Allg. nicht binär zu sein. In Bild 4.6.2b werden in einer ersten Stufe die Klassen $\Omega_1, \ldots, \Omega_k$ unterschieden, in der zweiten Stufe wird diese Klasseneinteilung weiter verfeinert in $\Omega_{11}, \ldots, \Omega_{kk_k}$. Es sind auch mehr als zwei Stufen denkbar, und die Art der verwendeten Merkmale wird i. Allg. von vorhergehenden Entscheidungen abhängen. Eine Verfeinerung der Klassen wird entweder heuristisch festgelegt oder über Hierarchien von Häufungsgebieten, worauf in Abschnitt 4.8.4 kurz eingegangen wird.

Aus Bild 4.6.3a geht hervor, dass ein binärer Klassifikationsbaum von der Art in Bild 4.6.2a immer dann aufgebaut werden kann, wenn der Merkmalsraum durch ein rechtwinkliges Gitter in Teilbereiche zerlegt wird. Aus Bild 4.6.3b geht hervor, dass ein derartiger Baum möglicherweise effektiver (mit weniger logischen Abfragen) darstellbar ist, und Bild 4.6.3c zeigt schließlich, dass auch beliebige rechteckige Gebiete auf Klassifikationsbäume zurückgeführt werden können. Die gestrichelten Linien deuten an, dass jede rechteckige Aufteilung trivialerweise zu einem rechtwinkligen Gitter erweitert werden kann, sodass Bäume wie in Bild 4.6.3c stets in

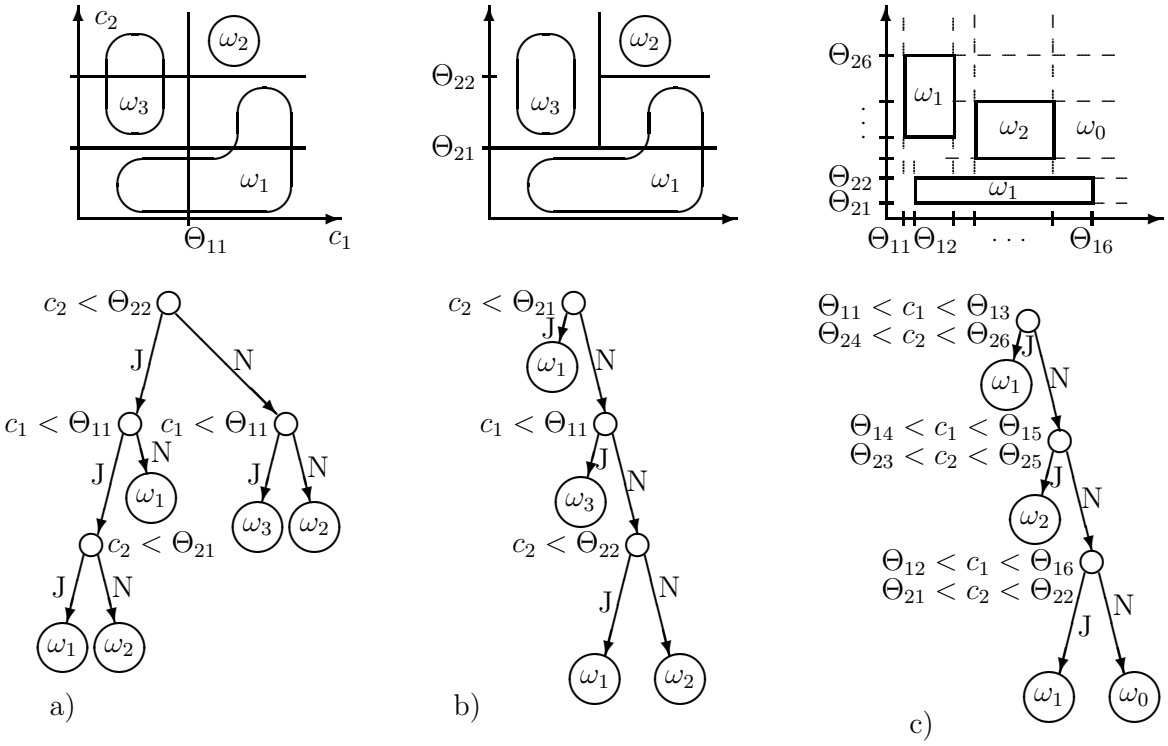


Bild 4.6.3: Bestimmte Aufteilungen des Merkmalsraumes führen zu einfachen Klassifikationsbäumen

binäre Bäume übergeführt werden können. Die Verwendung von Parallelen zu den Koordinatenachsen für die Aufteilung des Merkmalsraumes hat gegenüber anderen – wie in Bild 3.8.7, S. 240 – den Vorteil, dass nur einfache Schwellwertoperationen mit den Merkmalen erforderlich sind. Klassifikationsbäume lassen sich in der Weise verallgemeinern, dass in den Knoten beliebige logische Entscheidungen getroffen werden, z. B. von der Art „ist im linken oberen Bildviertel ein waagerechter Strich vorhanden“.

In der zitierten Literatur sind mehrere Vorschläge enthalten, wie für eine gegebene klassifizierte Stichprobe ω eine Zerlegung des Merkmalsraumes gemäß Bild 4.6.3a oder Bild 4.6.3c zu konstruieren ist. Darüberhinaus gibt es Algorithmen, um zu einem Baum wie in Bild 4.6.3a einen optimalen äquivalenten Baum zu konstruieren, beispielsweise einen Baum mit minimaler Anzahl von Knoten, in denen logische Entscheidungen zu treffen sind. Zusätzlich zu der rein intuitiv-heuristischen Erfindung eines Klassifikationsbaumes sind dieses Ansätze zu einer systematischen Konstruktion.

Wir beschränken uns hier auf die Angabe eines Algorithmus, der die Konstruktion rechteckiger Gebiete wie in Bild 4.6.3c erlaubt; er kann als ein **nichtparametrischer Klassifikator** angesehen werden. Das j-te Intervall auf der Koordinatenachse c_ν wird mit

$$I_{j\nu} = [a_{j\nu}, b_{j\nu}] \,, \quad a_{j\nu} \leq b_{j\nu} \tag{4.6.2}$$

bezeichnet. Die n Intervalle $I_{j\nu}$, $\nu = 1, \ldots, n$ definieren einen Hyperquader

$$E_j = I_{j1} \times I_{j2} \times \ldots \times I_{jn} \,, \tag{4.6.3}$$

Eingaben: $\omega_1,\ \omega_0 = \omega - \omega_1$; setze $j = 0$			
	IF	$j > 0$	
	THEN	$\omega_1 \leftarrow \omega_{\rm r}\ ,\quad \omega_{\rm r} \leftarrow \emptyset$	
	$j \leftarrow j+1\ ,\quad E_j \leftarrow {}^1\boldsymbol{c}_1,\ {}^1\boldsymbol{c}_1 \in \omega_1$		
	FOR alle Muster ${}^{\varrho}\boldsymbol{c}_1 \in \omega_1$:		
		IF	${}^{\varrho}\boldsymbol{c}_1 \notin E_j$
		THEN	IF $d({}^{k}\boldsymbol{c}_0, M(E_j,\ {}^{\varrho}\boldsymbol{c}_1)) \geq \theta_1$ für alle ${}^{k}\boldsymbol{c}_0 \in \omega_0$
			THEN ersetze $E_j \leftarrow M(E_j,\ {}^{\varrho}\boldsymbol{c}_1)$
			ELSE bringe ${}^{\varrho}\boldsymbol{c}_1$ in Menge $\omega_{\rm r}$
UNTIL $\omega_{\rm r} = \emptyset$			

Bild 4.6.4: Konstruktion vom $m+1 \leq N+1$ Hyperquadern

der als j-tes Ereignis bezeichnet wird. Die Verschmelzung zweier Ereignisse $E_j,\ j = 1, 2$ ergibt ein Ereignis E , das definiert ist mit

$$\begin{aligned} E &= M(E_1, E_2) \qquad (4.6.4)\\ &= I_1 \times I_2 \times \ldots \times I_n\ , \\ I_\nu &= [\min\{a_{1\nu}, a_{2\nu}\}, \max\{b_{1_\nu}, b_{2\nu}\}]\ . \qquad (4.6.5)\end{aligned}$$

Die Verschmelzung zweier Hyperquader E_1 und E_2 ist also der kleinste Hyperquader, der E_1 und E_2 enthält. Als Abstand eines Musters ${}^{\varrho}\boldsymbol{c}$ von einem Ereignis E wird definiert

$$\begin{aligned} d({}^{\varrho}\boldsymbol{c}, E) &= \sum_{\nu=1}^{n} \psi({}^{\varrho}c_\nu, E)\ , \qquad (4.6.6)\\ \psi({}^{\varrho}c_\nu, E) &= \begin{cases} 1 & : \ {}^{\varrho}c \notin I_\nu \\ 0 & : \ \text{sonst} \end{cases}\ . \end{aligned}$$

Eine Stichprobe ω enthalte N Muster und bestehe aus k Teilmengen $\omega_\kappa \subset \Omega_\kappa$ mit je N_κ Mustern. Die Stichprobe wird zerlegt in ω_1 und $\omega_0 = \omega - \omega_1$, wobei ω_0 nun $N_0 = N - N_1$ Muster enthält. Der Algorithmus in Bild 4.6.4 konstruiert Ereignisse, die alle Muster aus ω_1 und keine aus ω_0 enthalten. Im letzen Schritt des Algorithmus wurde zur Vereinfachung angenommen, dass Muster in ω_r erneut fortlaufend mit 1 beginnend indiziert werden. Das Prinzip des Algorithmus besteht darin, ein Ereignis so lange zu erweitern, wie ein Mindestabstand θ_1 zu Mustern aus ω_0 nicht unterschritten wird. Ein Muster, dessen Vereinigung mit einem Ereignis zu einer Unterschreitung des Abstandes führen würde, wird einem anderen Ereignis zugeordnet. Der Algorithmus ist k–mal für die Klassen $\omega_\kappa,\ \omega_0 = \omega - \omega_\kappa,\ \kappa = 1, \ldots, k$ auszuführen. Die Klassifikation erfolgt nach der Regel

$$\begin{aligned} &\text{bestimme } d(\boldsymbol{c}, E_{\kappa j}) \text{ für alle Klassen und alle Ereignisse ;}\\ &\text{wenn } \ d(\boldsymbol{c}, E_{\kappa j}) \leq \theta_2 \text{ für ein Ereignis genau einer Klasse } \Omega_\kappa\ , \qquad (4.6.7)\\ &\text{dann } \ \boldsymbol{c} \in \Omega_\kappa\ ,\ \text{sonst } \ \boldsymbol{c} \in \Omega_0\ . \end{aligned}$$

Man sollte $\theta_1 > \theta_2$ wählen, der Fall $\theta_2 = 0$ entspricht der Forderung, dass ein Muster in dem Hyperquader enthalten sein muss.

4.6.3 Klassifikator für nominale Merkmale

In der Literatur wird vielfach eine Unterscheidung zwischen ordinalen und nominalen Merkmalen getroffen, die etwa den hier verwendeten Begriffen numerisch und nichtnumerisch entsprechen. Bei einem nominalen Merkmal – z. B. dem Merkmal Farbe mit den Werten rot, grün, blau – lässt sich keine sinnvolle Ordnung nach der Größe oder dem Wert der Merkmale angeben, wie es z. B. bei dem Merkmal Fortmantfrequenz mit Werten $f_1 = 900$ Hz, $f_2 = 2000$ Hz, $f_3 = 2700$ Hz möglich ist. Daher ist auch bei nominalen Merkmalen die Angabe von Metriken oder Abständen und die Verwendung von darauf beruhenden Klassifikatoren nicht ohne weiteres möglich (s. jedoch (4.6.23)). In Abschnitt 3.1 wurde darauf verwiesen, dass sich in solchen Fällen die Interpretation der Merkmale als Symbolkette und die Verwendung syntaktischer Klassifikatoren anbieten. Es gibt jedoch auch andere Ansätze, von denen hier beispielhaft einer vorgestellt wird.

Ein grundsätzlicher Begriff bei der Klassifikation von Mustern mit nominalen Merkmalen ist das überdeckende Ereignis oder die überdeckende Symbolkette, was an einem einfachen Beispiel erläutert wird. Es wird angenommen, dass ein Muster nur mit den zwei Merkmalen Form und Farbe mit Werten Rechteck und Dreieck für die Form und rot, grün und blau für die Farbe dargestellt wird. Das spezielle Symbol β stehe für einen beliebigen Wert irgendeines Merkmals. Die Kette (Rechteck, β) überdeckt jede der Symbolketten (Rechteck, rot), (Rechteck, blau) und (Rechteck, grün). In Bild 4.6.5 wird ein Algorithmus angegeben, mit dem zu einer Stichprobe von Mustern mit nominalen Merkmalen eine als Definitionsmenge bezeichnete Menge ω_d von überdeckenden Ketten konstruiert wird. Die Definitionsmenge ist Grundlage der Klassifikation, die nach der Regel erfolgt

> ordne ein Muster der Klasse zu, deren Definitionsmenge eine Kette enthält, welche die Symbolkette des Musters überdeckt . (4.6.8)

Wenn also k Klassen vorliegen, sind k Definitionsmengen zu konstruieren.

Zur Angabe des Algorithmus bezeichnen wir den Wert des ν-ten Merkmals im j-ten Muster mit ${}^jc_\nu$. Das ν-te Merkmal c_ν nimmt Werte aus einer Menge S_ν an, wobei einige oder alle Mengen S_ν, $\nu = 1, \ldots, n$ gleich sein dürfen. Die Kette $(c_1, c_2, \ldots, c_n)$ *überdeckt* die Symbolkette $({}^jc_1, {}^jc_2, \ldots, {}^jc_n)$ eines Musters wenn entweder $c_\nu = {}^jc_\nu$ oder $c_\nu = \beta$, $\nu = 1, \ldots, n$ gilt. Die Stichprobe ω sei wie in Abschnitt 4.6.2 in ω_1 und $\omega_0 = \omega - \omega_1$ zerlegt. Am Ende enthält $\omega_1 = \omega_\mathrm{d}$ eine Definitionsmenge für ω_1. Im Algorithmus in Bild 4.6.5 wird die Verschmelzung von Ketten so lange wiederholt, bis keine neuen überdeckenden Ketten mehr gebildet werden; dann ist $\omega_1 = \omega_\mathrm{d}$. Die Klassifikation erfolgt gemäß (4.6.8).

4.6.4 Abstandsmessende Klassifikatoren

Vorgehensweisen

Die Verwendung eines geeigneten Abstandsmaßes zwischen einem **Prototypen** oder *Referenzmuster* (oder einer Schablone, “template”) und einem neu beobachteten Muster oder *Testmuster* ist ein intuitiv nahe liegender Ansatz bei der Klassifikation. Für solche *abstandsmessenden Klassifikatoren* gibt es eine Reihe überwiegend heuristisch motivierter Varianten, von denen einige als Beispiele genannt werden:

1. In Abschnitt 4.2.5 ergab sich als Spezialfall des statistischen Klassifikators ein bewichteter Abstand des Merkmalsvektors eines neuen Musters vom Klassenzentrum μ_λ.

definiere die Verschmelzung zweier Symbolketten $({}^{i}c_1, {}^{i}c_2, \ldots, {}^{i}c_n)$ und $({}^{j}c_1, {}^{j}c_2, \ldots, {}^{j}c_n)$ mit

$$(c_1, c_2, \ldots, c_n)_{ij} = ({}^{i}c_1, {}^{i}c_2, \ldots, {}^{i}c_n) \cup ({}^{j}c_1, {}^{j}c_2, \ldots, {}^{j}c_n)$$

$$c_\nu = \begin{cases} \beta & : \ {}^{i}c_\nu \neq {}^{j}c_\nu \\ {}^{i}c_\nu & : \ {}^{i}c_\nu = {}^{j}c_\nu \end{cases}$$

bilde die Verschmelzung aller Paare von Symbolketten aus ω_1	
IF	das Ergebnis der Verschmelzung eines Paares aus ω_1 überdeckt keine Kette aus ω_0
THEN	ordne dieses Ergebnis der Menge ω_d zu
bringe auch alle Symbolketten nach ω_d, bei denen das Ergebnis der Verschmelzung mit irgendeiner Symbolkette nicht nach ω_d gebracht wurde	
IF	$\omega_\mathrm{d} \neq \omega_1$
THEN	ersetze $\omega_1 \leftarrow \omega_\mathrm{d}$, $\omega_\mathrm{d} \leftarrow \emptyset$

UNTIL $\omega_\mathrm{d} = \omega_1$

Bild 4.6.5: Zur Konstruktion einer Definitionsmenge ω_d

2. Eine einfache (und i. Allg. sehr suboptimale) Heuristik besteht in der Berechnung des EUKLID-Abstands zum Mittelwert der Stichprobe der Klasse Ω_λ.
3. Die NN–Regel in Abschnitt 4.2.7 beruht auf dem Vergleich von Abständen zu allen Mustern aus der Trainingsstichprobe.
4. Mit der Hauptachsentransformation in Abschnitt 3.8.2 wird eine klassenspezifische Transformationsmatrix je Klasse berechnet, ein neues Muster nach allen k klassenspezifischen Matrizen entwickelt und mit den Entwicklungskoeffizienten rekonstruiert. Die Rekonstruktion mit dem kleinsten Abstand (z. B. dem kleinsten mittleren quadratischen Abstand) zum Original bestimmt die Klasse des neuen Musters.
5. Die in Abschnitt 3.5.3 erörterten Merkmalsfilter, die dort als spezielle lineare Systeme eingeführt wurden, berechnen ebenfalls ein Abstandsmaß, wie aus (4.6.9) – (4.6.11) hervorgeht.
6. Mit der unten erläuterten dynamischen Programmierung wird eine optimale *nichtlineare* Verzerrung zwischen Referenz– und Testmuster berechnet, sodass der Abstand minimiert wird.

Ist $[f_{\lambda jk}]$ das Referenzmuster der Klasse Ω_λ und $[f_{jk}]$ ein Testmuster, so ist der mittlere quadratische Fehler

$$D_\lambda = \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} (f_{\lambda jk} - f_{jk})^2 \tag{4.6.9}$$

ein mögliches Abstandsmaß. Wenn k Klassen vorliegen, entscheidet man sich für die mit dem kleinsten Fehler oder dem kleinsten Abstand zum Testmuster. Nun ist aber die Lage des Minimums von

$$D_\lambda = \sum\sum f_{\lambda jk}^2 + \sum\sum f_{jk}^2 - 2\sum\sum f_{\lambda jk} f_{jk} \tag{4.6.10}$$

unabhängig von $\sum\sum f_{jk}^2$, und wenn alle Referenzmuster $\boldsymbol{f}_\lambda$ auf den Wert $\sum\sum f_{\lambda jk}^2 = 1$ normiert sind, ist diese Lage auch unabhängig von $\sum\sum f_{\lambda jk}^2$. Der Abstand D_λ ist also dann klein, wenn der Term $\sum\sum f_{\lambda jk} f_{jk}$ groß ist. Das entspricht – bis auf eine Umindizierung und die Verwendung fester Indizes $j_0 = k_0 = 0$ – der Beziehung (3.5.24), S. 205. Man bezeichnet

$$R_{\lambda jk} = \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{M-1} f_{\lambda,j+\mu,k+\nu}\, f_{\mu\nu} \tag{4.6.11}$$

als **Kreuzkorrelation** zwischen $\boldsymbol{f}_\lambda$ und $\boldsymbol{f}$. Die obigen Ausführungen zeigen, dass ein kleiner Abstand (4.6.9) einer großen Korrelation (4.6.11) für $j = k = 0$ entspricht.

Es ist eine naheliegende Verallgemeinerung, im Bedarfsfalle statt eines Referenzmusters je Klasse mehrere zu verwenden. Ebenso ist es für das Prinzip der abstandsmessenden Klassifikatoren belanglos, ob man die Abtastwerte vergleicht oder irgendwelche daraus abgeleitete Größen, wie etwa die FOURIER-Koeffizienten oder sonstige Merkmale aus Kapitel 3; dieses soll durch die Bezeichnungen $\boldsymbol{f}_\lambda$ und $\boldsymbol{f}$ nicht ausgeschlossen werden. Schließlich gibt es außer (4.6.9) noch andere Abstandsmaße wie (4.2.146), S. 355, von denen insbesondere noch

$$D_\lambda = \sum\sum \mid f_{\lambda jk} - f_{jk} \mid \tag{4.6.12}$$

wegen der einfachen Berechenbarkeit praktisch interessant ist.

Nichtlineare Verzerrungen

Ein Problem besteht darin, dass Verzerrungen von $\boldsymbol{f}$ sich auf die Abstandsberechnung in vollem Umfang auswirken, Bild 4.6.6 zeigt dafür ein Beispiel. Vom subjektiven Eindruck her sieht das Testmuster dem Referenzmuster 1 „recht ähnlich“, aber nicht dem Referenzmuster 2. Trotzdem liefert (4.6.12) für beide den gleichen Abstand, nämlich 132; die Muster wurden einfach so konstruiert, dass je Komponente die Betragsdifferenz gleich ist. Eine lineare Normierung der Länge des Testmusters gemäß Abschnitt 2.5.3 ergibt $D_1 = 79$ und $D_2 = 275$. Tatsächlich ist dadurch der Abstand zur Referenz 1 deutlich kleiner geworden als zur Referenz 2, sodass eine Klassifikation möglich ist. Trotzdem ist die lineare Normierung hier unbefriedigend. Eine genauere Betrachtung des Testmusters zeigt nämlich, dass dieses gegenüber dem Referenzmuster 1 *nichtlinear* verzerrt wurde. Derartige nichtlineare Verzerrungen treten beispielsweise in gesprochenen Wörtern auf, da verschiedene Laute von verschiedenen Sprechern in unterschiedlicher Weise gedehnt werden können. Zum Beispiel kann in dem Wort „Bote“ der Vokal „o“ in einem relativ großen Bereich in seiner Länge schwanken, der Plosivlaut „t“ dagegen nur in einem wesentlich kleineren. Bei der Normierung müssen daher unterschiedliche Zeitabschnitte in unterschiedlicher Form abgebildet werden. Das Prinzip dafür zeigt Bild 4.6.7; es beruht darauf, dass Normierung (oder nichtlineare Abbildung) und Klassifikation (oder Abstandsberechnung) *kombiniert* werden. Das formale Hilfsmittel dafür ist die *dynamische Programmierung* (DP), deren Prinzip in Abschnitt 1.6.8 vorgestellt wurde.

Zur Definition der Eigenschaften der gesuchten nichtlinearen Abbildung werden in Bild 4.6.7 zwei eindimensionale Folgen $[f_{\lambda j}]$, $j = 0, 1, \ldots, M_\lambda - 1$ und $[f_k]$, $k = 0, 1, \ldots, M - 1$ als Referenz- und Testmuster verwendet. Die Abbildung wird durch eine diskrete **Verzerrungsfunktion** (“warping function”)

$$k = w(j)\ , \quad \text{mit } 0 = w(0)\ , \quad M - 1 = w(M_\lambda - 1) \tag{4.6.13}$$

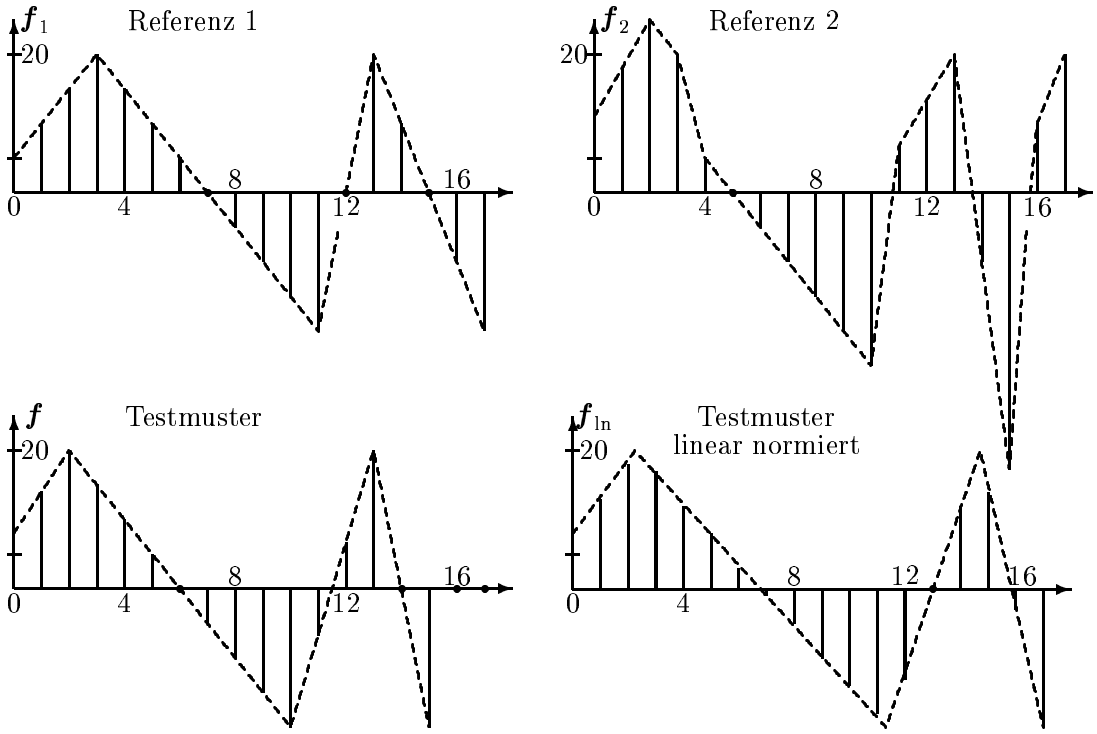


Bild 4.6.6: Das Testmuster hat zu beiden Referenzmustern den *gleichen* Abstand nach (4.6.12); durch eine lineare Normierung der Dauer erniedrigt sich der Abstand zwischen Test– und Referenzmuster 1 von 132 auf 79, der zwischen Test– und Referenzmuster 2 steigt von 132 auf 275

in der (j,k)-Ebene definiert. Die M_W Wertepaare $(j,\ k = w(j))$ denke man sich in der Reihenfolge, in der sie beim Durchlaufen der Kurve von links unten nach rechts oben auftreten, nach einem Laufindex l geordnet. Die Punkte auf der Kurve, die die Abbildung definiert, sind dann durch die Indizes $(j(l), k(l))$, $l = 1, \ldots, M_W$ gegeben. Jede Kurve $k = w(j)$ definiert also auch einen *Pfad* in der (j,k)-Ebene. Als Abstand der Muster kann man wie in (4.6.12)

$$D_\lambda = \sum_{l=1}^{M_W} \mid f_{\lambda j(l)} - f_{k(l)} \mid \tag{4.6.14}$$

wählen oder mit irgendeinem geeigneten Abstandsmaß $d(f_{\lambda j}, f_k) \geq 0$ ein Maß

$$D_\lambda = \sum_{l=1}^{M_W} d(f_{\lambda j(l)}, f_{k(l)}) \ , \tag{4.6.15}$$

dessen Wert von der gewählten Verzerrungsfunktion abhängt. Die optimale Verzerrungsfunktion w^* ist diejenige, welche D_λ minimiert

$$D_\lambda^* = D_\lambda(w^*) = \min_w D_\lambda \ . \tag{4.6.16}$$

Zum Beispiel ergibt die Verzerrungsfunktion in Bild 4.6.7 einen Wert $D_1^* = 41$, d. h. durch die nichtlineare Abbildung wird der Abstand zwischen Test- und Referenzmuster gegenüber der linearen Normierung deutlich verringert. Aus (4.6.16), (4.6.15) geht hervor, dass die nichtlineare

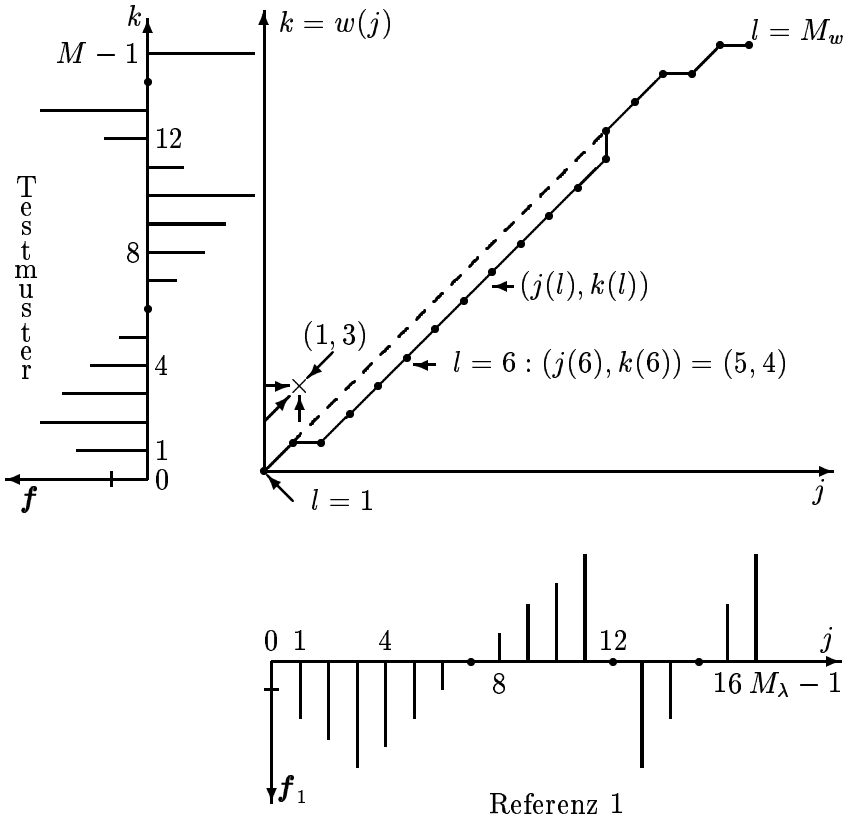


Bild 4.6.7: Durch eine nichtlineare Normierung der Dauer erniedrigt sich der Abstand zwischen Test– und Referenzmuster 1 von 132 auf 41

Abbildung $k = w^*(j)$ bezüglich eines Referenzmusters $\boldsymbol{f}_\lambda$ definiert wird; in diesem Sinne sind Normierung und Klassifizierung kombiniert, was bei der linearen Normierung nicht erforderlich und nicht möglich ist.

Berechnung der optimalen Verzerrungsfunktion

Es bleibt nun noch die tatsächliche Berechnung von w^*. Praktisch wird es ausreichen, die Menge der zulässigen Funktionen $w(j)$ auf einen Teilbereich der (j,k)-Ebene zu beschränken. Zum Beispiel wird in der Sprachverarbeitung oft die Anforderung gestellt, dass

$$w(j+1) - w(j) = \begin{cases} 0,1,2 & : \quad w(j) \neq w(j-1) \\ 1,2 & : \quad w(j) = w(j-1) \end{cases} \quad . \tag{4.6.17}$$

Damit wird die Steigung von $w(j)$ beschränkt und insbesondere ein senkrechter Anstieg ausgeschlossen. Eine andere Möglichkeit besteht darin, die Indexpaare $(j(l), k(l))$ zu beschränken auf die drei Paare

$$(j(l), k(l)) = \begin{cases} (j(l-1),\ k(l-1)+1) \\ (j(l-1)+1,\ k(l-1)+1) \\ (j(l-1)+1,\ k(l-1)) \end{cases} \quad . \tag{4.6.18}$$

Die *Vorgänger* eines Punktes $(j(l), k(l))$ können also links neben, senkrecht unter oder unter $45°$ nach links unten liegen. Die Berechnung von w^* erfolgt, wie schon gesagt, mit der

dynamischen Programmierung (DP), bei der eine vollständige Suche über alle zulässigen Funktionen $k = w(j)$ ausgeführt wird. Der für l Wertepaare $f_{\lambda j}, f_k$ berechnete Abstand D_λ^l mit $1 \leq l \leq M_W$ in (4.6.15) bleibt unverändert, wenn ein $(l+1)$-tes Wertepaar hinzukommt, sodass für den mit $l+1$ Wertepaaren berechneten Abstand D_λ^{l+1} gilt

$$D_\lambda^{l+1} = D_\lambda^l + d(f_{\lambda j(l+1)}, f_{k(l+1)}) \; . \tag{4.6.19}$$

Offensichtlich ist dieses eine monotone und separierbare Gütefunktion wie sie für die DP erforderlich ist. Hat man also zu irgendeinem Punkt $(j(l), k(l))$ der (j,k)-Ebene einen optimalen Pfad gefunden – d. h. einen Pfad, auf dem der Abstand minimiert wird – so muss wegen des *Optimalitätsprinzips* jeder andere optimale Pfad zu einem Punkt $(j(l+1), k(l+1))$, der auch $(j(l), k(l))$ enthalten soll, den optimalen Pfad nach $(j(l), k(l))$ enthalten. Daraus folgt, dass man bei der Suche der besten Funktion w^* nicht alle möglichen Pfade, die zu einem Punkt führen, speichern muss, sondern nur den jeweils besten. Der dynamischen Programmierung liegt die Anwendung dieses Prinzips zugrunde, die zu einer ganz wesentlichen Reduktion des Speicher- und Rechenaufwandes führt, wie in Abschnitt 1.6.8 erläutert. Wenn die zulässigen Vorgänger zunächst nicht eingeschränkt sind, ergibt sich damit die Rekursionsgleichung

$$\boxed{D^*(j,k) = d(f_{\lambda j}, f_k) + \min\{\text{mögliche Vorgänger}\} \; .} \tag{4.6.20}$$

Als Beispiel wird die Berechnung von w^* unter Berücksichtigung von (4.6.18) angegeben. Mit den Bezeichnungen von Bild 4.6.7 erfolgt die Berechnung spaltenweise, d. h. für festes j werden alle Abstände zu Punkten (j,k), $k = 0, \ldots, M-1$ berechnet und dann j um 1 erhöht. Mit $D^*(j,k)$ wird der minimale Abstand bezeichnet, wenn man die Werte $0, 1, \ldots, j$ von $\boldsymbol{f}_\lambda$ mit den Werten $0, 1, \ldots, k$ von $\boldsymbol{f}$ vergleicht. Es ist wichtig, dass $D^*(j,k)$ der minimale Abstand ist, den man mit der optimalen Funktion w^* erreicht. In der ersten Spalte $(j = 0)$ ist jeder Punkt $(j = 0, k)$ nur auf einem Weg erreichbar, also jeder Abstand bereits minimal. In der zweiten Spalte kann z. B. der Punkt $(j = 1, k = 3)$ in einem Schritt wegen (4.6.18) von den drei Vorgängern $(j = 1, k = 2), (j = 0, k = 2), (j = 0, k = 3)$ erreicht werden. Es gilt also für den minimalen Abstand $D^*(j,k)$ die rekursive Beziehung

$$D^*(j,k) = d(f_{\lambda j}, f_k) + \min\left\{D^*(j, k-1),\; D^*(j-1, k-1),\; D^*(j-1, k)\right\} \; . \tag{4.6.21}$$

Es gibt hier also nur drei möglichen Vorgänger. Die Gleichung (4.6.21) wird für die gesamte (j,k)-Ebene ausgewertet bis man mit $D^*(M_\lambda - 1, M-1)$ den gesuchten minimalen Abstand zwischen $\boldsymbol{f}_\lambda$ und $\boldsymbol{f}$ erhält. Wenn der Punkt (j,k) von mehr oder anderen als den oben genannten drei Vorgängern erreichbar ist, ändert sich in (4.6.21) lediglich die Menge der Abstände, über die das Minimum gesucht wird. Wenn man nach Erreichen von $j = M_\lambda - 1$ und $k = M-1$ den optimalen Pfad angeben will, muss bei jeder Auswertung von (4.6.21) nicht nur $D^*(j,k)$ gespeichert werden, sondern auch die Indizes des auf dem optimalen Pfad nach (j,k) liegenden Vorgängers. Eine Beschränkung des Bereiches der (j,k)-Ebene, in der der optimale Pfad oder w^* liegen darf, erhält man, indem in jeder Spalte der Index k nicht von 0 bis $M-1$, sondern nur über einen geeignet gewählten Bereich variiert wird. Zusammengefasst ergibt sich der in Bild 4.6.8 gezeigte Algorithmus.

Erweiterungen

Klassifikatoren, die auf einer nichtlinearen Normierung mit Hilfe der dynamischen Programmierung beruhen, haben insbesondere im Bereich der Spracherkennung eine große Bedeutung

FOR $j = 0$ TO $M_\lambda - 1$ (d.h. für alle Spalten)		
	IF	der Suchbereich in der (j,k)-Ebene ist eingeschränkt
	THEN	bestimme die untere Grenze $K_u(j)$ und die obere Grenze $K_o(j)$ des Index k
	ELSE	setze $K_u(j) = 0$, $K_o(j) = M - 1$
	FOR $k = K_u(j)$ TO $K_o(j)$ (d.h. für alle Punkte einer Spalte)	
		berechne $D^*(j,k) = d(f_{\lambda j}, f_k) + \min\{D^*(j,k-1), D^*(j-1,k-1), D^*(j-1,k)\}$; setze $D^*(j,k) = 0$ für $j < 0$ oder $k < 0$ (Randbereich der (j,k)-Ebene)
		IF optimaler Pfad soll rekonstruiert werden
		THEN speichere die Indizes des Vorgängers auf dem optimalen Pfad
der minimale Abstand ist $D^*(M_\lambda - 1, M - 1)$, der optimale Pfad kann bei Bedarf von $j = M_\lambda - 1$, $k = M - 1$ aus über die Indizes der Vorgänger rekonsturiert werden		

Bild 4.6.8: Berechnung der besten Verzerrungsfunktion

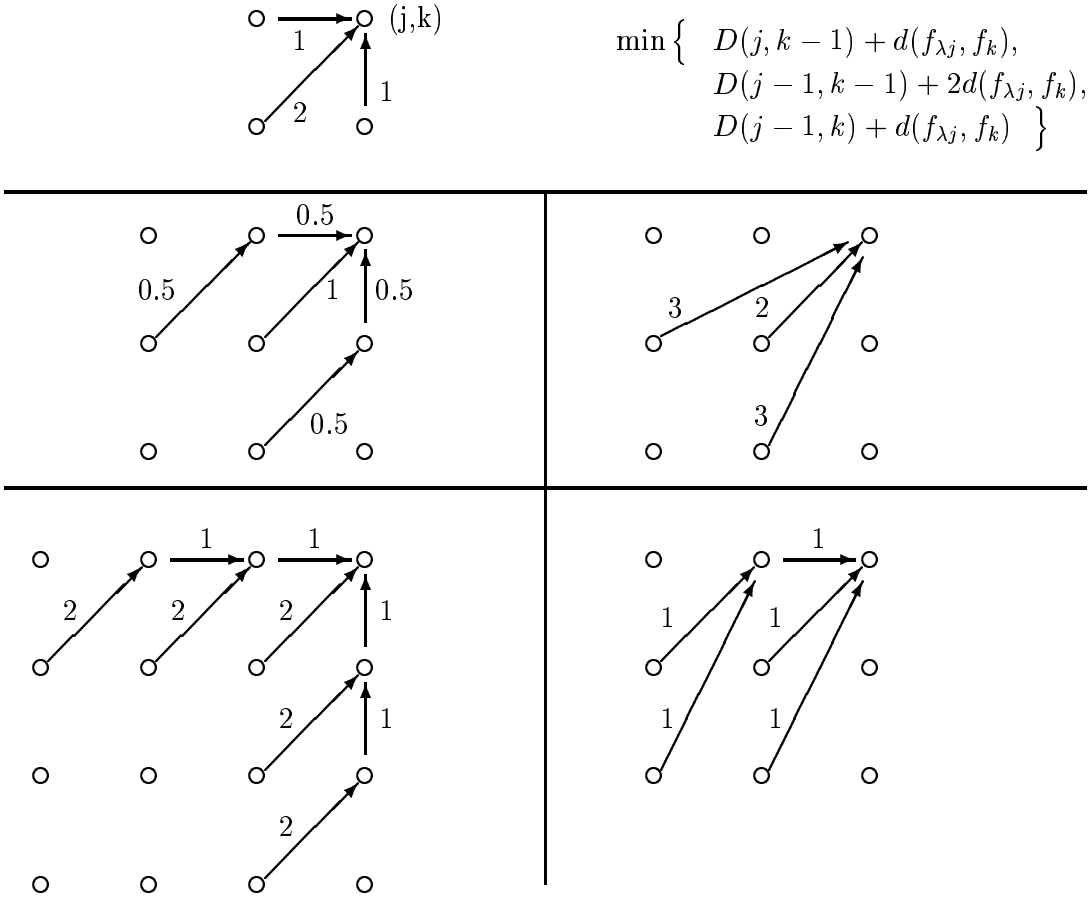


Bild 4.6.9: Weitere Beispiele für die Definition von möglichen Vorgängern

erlangt. In diesem Zusammenhang sind auch andere Abstandsmaße als (4.6.9), (4.6.12) entwickelt worden. Ebenso gibt es Modifikationen von D_λ in (4.6.15), z. B.

$$D_\lambda \;=\; \sum_{l=1}^{M_w} d(f_{\lambda j(l)}, f_{k(l)}) \frac{\alpha(l)}{N(\alpha)} \;,$$

$$\begin{aligned} \alpha(l) &= j(l) - j(l-1) + k(l) - k(l-1) \ , \\ N(\alpha) &= \sum_{l=1}^{M_w} \alpha(l) \ , \\ N(\alpha) &= M + M_\lambda \quad \text{für angegebenes } \alpha(l) \ . \end{aligned} \tag{4.6.22}$$

Dabei ist $\alpha(l)$ eine Gewichtung der Abstände $d(f_{\lambda j(l)}, f_{k(l)})$ und $N(\alpha)$ eine Normierung des Abstandes D_λ. Weitere Beispiele für die Definition von Vorgängern zeigt Bild 4.6.9.

Schließlich ist zu erwähnen, dass es sich bei $d(f_{\lambda j(l)}, f_{k(l)})$ in (4.6.15) im Prinzip um *irgendein* Abstandsmaß handeln darf, also z. B. auch um einen geeignet definierten Abstand zwischen zwei Symbolketten $v_{\lambda j},\ v_j,\ j = 0, 1, , \ldots, n$. Ein Beispiel ist der LEVENSTEIN-*Abstand*

$$\begin{aligned} D_\lambda &= \sum_{j=0}^{n} d(v_{\lambda j}, v_j) \ , \\ d(v_{\lambda j}, v_j) &= \begin{cases} 1 & : \ v_{\lambda j} \neq v_j \\ 0 & : \ v_{\lambda j} = v_j \end{cases} \ . \end{aligned} \tag{4.6.23}$$

Damit wird die nichtlineare Abbildung und Abstandsberechnung auch auf den Abstand zweier Symbolketten verallgemeinert. Mit der Einschränkung (4.6.18) ergibt sich dann als Abstand zweier Symbolketten die kleinste Anzahl von Einfügungen, Löschungen und Vertauschungen von Symbolen, die erforderlich ist, um die eine Kette in die andere zu überführen.

4.7 Klassifikation im Kontext

Bereits in Abschnitt 1.5 wurde die Berücksichtigung des Kontext bei der Klassifikation eingeführt. Die beiden Beispiele in Bild 4.7.1 zeigen, dass das gleiche Schriftzeichen je nach **Kontext**, d. h. je nach den benachbarten Zeichen, unterschiedlich beurteilt werden kann. Ein Klassifikator, der jeweils nur ein Zeichen angeboten bekommt und dieses *unabhängig* von allen anderen klassifiziert, wird in unsicheren Fällen dieses Zeichen entweder zurückweisen oder falsch klassifizieren. Es ist aber offensichtlich, dass in vielen Fällen durch die Verwendung von Kontext auch unsichere Zeichen noch richtig klassifizierbar sind. Zwar wurde in Abschnitt 1.2 Klassifikation von einfachen Mustern als von anderen Mustern unabhängig zu lösende Aufgabe definiert und in Abschnitt 1.4 das Thema des Buches auf die Klassifikationsaufgabe eingeschränkt; aber die Berücksichtigung von Kontext ist eine so naheliegende Erweiterung und bringt nach den veröffentlichten Ergebnissen einen so wichtigen Beitrag, dass zumindest eine kurze Erörterung hier angemessen ist. Im Folgenden wird auf verschiedene Verfahren kurz eingegangen und stets die Vorstellung zugrunde gelegt, dass eine Folge

$$\boldsymbol{F} = ({}^1\boldsymbol{f}, {}^2\boldsymbol{f}, \ldots, {}^N\boldsymbol{f}) \tag{4.7.1}$$

von Einzelmustern ${}^\varrho\boldsymbol{f}$ beobachtet wird. Der Folge von Mustern wird eine Folge von Klassennamen

$$\boldsymbol{\Omega} = ({}^1\Omega, {}^2\Omega, \ldots, {}^N\Omega) \;\text{ mit }\; {}^i\Omega \in \{\Omega_1, \Omega_2, \ldots, \Omega_k\} \tag{4.7.2}$$

zugeordnet. Gesucht ist die „richtige“ Folge $\boldsymbol{\Omega}$ unter Berücksichtigung *aller* N getroffenen Entscheidungen. Beispiele für Folgen $\boldsymbol{F}$ sind eine Folge von Buchstaben eines geschriebenen Wortes, von Lauten eines gesprochenen Wortes oder von Bildpunkten eines Multispektralbildes.

Statistischer Ansatz

Es ist naheliegend, den BAYES-*Klassifikator* mit der Entscheidungsregel (4.1.33) zu verallgemeinern und auf das obige Problem anzuwenden. Aus den Mustern der Folge $\boldsymbol{F}$ werden n-dimensionale Merkmalsvektoren ${}^j\boldsymbol{c}$ extrahiert und zu einem Vektor

$$\boldsymbol{C} = ({}^1\boldsymbol{c}^\mathsf{T}, {}^2\boldsymbol{c}^\mathsf{T}, \ldots, {}^N\boldsymbol{c}^\mathsf{T})^\mathsf{T} \tag{4.7.3}$$

zusammengesetzt. Die a priori Wahrscheinlichkeit dafür, dass $\boldsymbol{\Omega}$ eine bestimmte Folge von Werten (Klassen) annimmt, wird mit $p(\boldsymbol{\Omega})$ bezeichnet. Wie schon in (4.1.42), S. 318, ausgeführt, ist in direkter Verallgemeinerung von (4.1.33), (4.1.34) die a posteriori Wahrscheinlichkeit von $\boldsymbol{\Omega}$ nach Beobachtung von $\boldsymbol{C}$

$$p(\boldsymbol{\Omega}\,|\,\boldsymbol{C}) = \frac{p(\boldsymbol{\Omega})\,p(\boldsymbol{C}\,|\,\boldsymbol{\Omega})}{p(\boldsymbol{C})}\;. \tag{4.7.4}$$

Die Klassifikation erfolgt gemäß Satz 4.3 in Abschnitt 4.1.4 Statt der Bestimmung des Maximums von $p(\boldsymbol{\Omega}\,|\,\boldsymbol{C})$ genügt auch die Bestimmung des Maximums von $p(\boldsymbol{\Omega})\,p(\boldsymbol{C}\,|\,\boldsymbol{\Omega})$ bezüglich der möglichen Werte von $\boldsymbol{\Omega}$. Auch die Anwendung des in Abschnitt 4.1.3 getroffenen allgemeineren Ansatzes ist im Prinzip möglich. Es ist offensichtlich, dass die Anwendung von (4.7.4) in dieser Form in der Regel am Aufwand scheitern wird, da die Werte von $p(\boldsymbol{\Omega})$ und die Dichten $p(\boldsymbol{C}\,|\,\boldsymbol{\Omega})$ für alle möglichen Werte von $\boldsymbol{\Omega}$ bestimmt und gespeichert werden müssen. Mit den

Bild 4.7.1: Der Einfluss von Kontext auf die Zeichenerkennung

Bezeichnungen von (4.7.2) sind dieses k^N Werte und k^N Dichten von $n \cdot N$–dimensionalen Vektoren.

Zur Anwendung von (4.7.4) sind vereinfachende Annahmen erforderlich. Eine mögliche besteht darin, statistische Unabhängigkeit der Merkmalsvektoren anzunehmen. Dann ist

$$p(\boldsymbol{C}\,|\,\boldsymbol{\Omega}) = \prod_{\varrho=1}^{N} p({}^{\varrho}\boldsymbol{c}\,|\,{}^{\varrho}\Omega = \Omega_\kappa)\ . \tag{4.7.5}$$

Die Annahme statistischer Unabhängigkeit der Werte von ${}^1\Omega, \ldots, {}^N\Omega$ würde dagegen den Verzicht auf Kontextberücksichtigung bedeuten und ist daher nicht möglich. Es gilt hier

$$\begin{aligned} p(\boldsymbol{\Omega}) &= p({}^1\Omega, {}^2\Omega, \ldots, {}^N\Omega) \\ &= p({}^1\Omega)\, p_u({}^2\Omega\,|\,{}^1\Omega)\, p_u({}^3\Omega\,|\,{}^1\Omega, {}^2\Omega) \ldots p_u({}^N\Omega\,|\,{}^1\Omega\ \ldots\ {}^{N-1}\Omega)\ . \end{aligned} \tag{4.7.6}$$

Eine vereinfachende Annahme kann nun darin bestehen, dass man nur Abhängigkeiten zum direkten Nachfolger berücksichtigt. Dann ist

$$p(\boldsymbol{\Omega}) = p({}^1\Omega)\, p_u({}^2\Omega\,|\,{}^1\Omega)\, p_u({}^3\Omega\,|\,{}^2\Omega) \ldots p_u({}^N\Omega\,|\,{}^{N-1}\Omega)\ . \tag{4.7.7}$$

Statt k^N Werten sind nur k^2 Werte $p_u(\Omega_\kappa\,|\,\Omega_\lambda),\ \ \kappa, \lambda = 1, \ldots, k$ und die k Werte $p(\Omega_\kappa)$ zu speichern. Mit $p_u(\Omega_\kappa\,|\,\Omega_\lambda)$ wird die Wahrscheinlichkeit bezeichnet, dass ein Muster aus Ω_κ auftritt, wenn direkt vorher eines aus Ω_λ aufgetreten ist. Diese **Übergangswahrscheinlichkeiten** lassen sich durch Auszählen von Paaren schätzen; sie sind nicht mit den bedingten Fehlerwahrscheinlichkeiten $p(\Omega_\lambda\,|\,\Omega_\kappa)$ in (4.1.8) zu verwechseln. Aus Bild 1.5.2, S. 31 geht hervor, dass die effiziente Berechnung der maximalen a posteriori Wahrscheinlichkeit von der effizienten *Suche* unter den möglichen Alternativen abhängt. Dieses ist mit dem VITERBI-Algorithmus möglich.

VITERBI-Algorithmus

Es handelt sich hierbei um einen Algorithmus, der mit Hilfe der dynamischen Programmierung (DP) die Folge $\boldsymbol{\Omega}$ mit maximaler a posteriori Wahrscheinlichkeit bestimmt, wenn man die Kosten bzw. die Gewichte der Kanten geeignet wählt. Im Prinzip ist das auch das Anliegen in (4.7.4) und Satz 4.3.

Die N Merkmalsvektoren ${}^{\varrho}\boldsymbol{c}$ und die k je Vektor möglichen Klassen Ω_κ sind in Bild 4.7.2 als Knoten eines Netzwerkes gezeigt. Der zum Vektor ${}^{\varrho}\boldsymbol{c}$ und zur Klasse Ω_κ gehörige Knoten entspricht dem Ereignis, dass der Vektor ${}^{\varrho}\boldsymbol{c}$ beobachtet wird, wenn das Muster aus Ω_κ ist; ihm wird das Gewicht $\ln p({}^{\varrho}\boldsymbol{c}\,|\,\Omega_\kappa)$ zugeordnet. Die den Knoten zugeordneten Gewichte sind also abhängig von den Beobachtungen. Jede Kante zwischen zwei Knoten, von denen der eine zu ${}^{\varrho}\boldsymbol{c}$ und Ω_κ, der andere zu ${}^{\varrho-1}\boldsymbol{c}$ und Ω_λ gehört, entspricht dem Ereignis, dass auf ein Muster aus

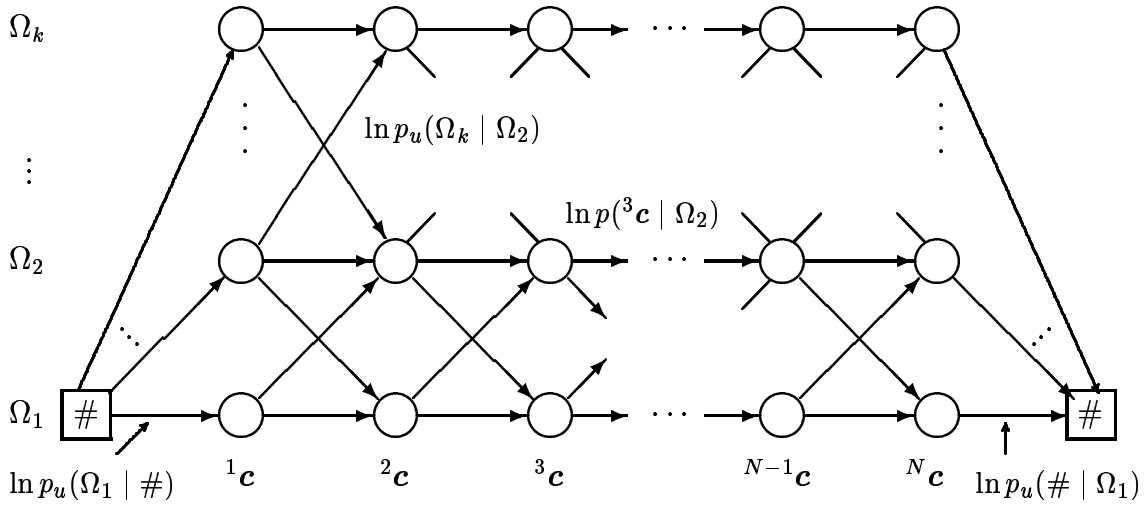


Bild 4.7.2: Mit dem VITERBI-Algorithmus (bzw. der DP) wird der Pfad mit dem größten Gewicht durch das Netzwerk gefunden

Ω_λ eines aus Ω_κ folgt; der Kante wird das Gewicht $\ln p_u(\Omega_\kappa|\Omega_\lambda)$ zugeordnet, das unabhängig von den beobachteten Merkmalsvektoren ist. Wenn man annimmt, dass die Folge $\boldsymbol{F}$ in (4.7.1) links und rechts durch ein spezielles Symbol # begrenzt ist, lassen sich auch den Anfangs- und Endkanten Übergangswahrscheinlichkeiten und damit Gewichte zuordnen. Jede Folge $\boldsymbol{\Omega}$ von Klassen, die einer Folge $\boldsymbol{F}$ mit Merkmalsvektoren $\boldsymbol{C}$ zugeordnet wird, entspricht einem Pfad durch das Netzwerk in Bild 4.7.2 vom Anfangs- zum Endknoten. Das Gewicht dieses Pfades ist die Summe der Kanten- und Knotengewichte. Der beste Pfad durch das Netz (und damit auch die beste Zuordnung $\boldsymbol{\Omega}$ von Klassen zum Vektor $\boldsymbol{C}$) ist der mit größtem Gewicht. Setzt man $p_u(\Omega_\kappa|\#) = p(\Omega_\kappa)$ und $p_u(\#|\Omega_\kappa) = 1/k$, so entspricht in diesem Fall der Pfad maximalen Gewichts der Folge $\boldsymbol{\Omega}$ mit der größten a posteriori Wahrscheinlichkeit in (4.7.4). Das ergibt sich unmittelbar, wenn man (4.7.5) und (4.7.7) in (4.7.4) einsetzt und statt $p(\boldsymbol{\Omega}|\boldsymbol{C})$ die bezüglich der Lage des Maximums äquivalente Größe $\ln p(\boldsymbol{\Omega}|\boldsymbol{C})$ betrachtet.

Der Pfad maximalen Gewichtes wird wie in Bild 4.7.3 angegeben ermittelt. Die Vorgehensweise ist ähnlich wie in Abschnitt 4.6.4. Der VITERBI Algorithmus wird sowohl in der Spracherkennung als auch bei der Korrektur von Zeichenketten eingesetzt. Das Prinzip ist stets das in Bild 4.7.3 angegebene, jedoch werden natürlich im Einzelfalle entsprechende Modifikationen vorgenommen.

Wörterbuch

Unter einem **Wörterbuch** ist hier allgemein die Menge $\widetilde{\boldsymbol{\Omega}}$ der zulässigen oder gültigen Folgen $\boldsymbol{\Omega}$ zu verstehen, wobei die Länge N der Folge $\boldsymbol{\Omega}$ fest oder in bestimmten Grenzen variabel sein kann. Sind zum Beispiel die beobachteten Muster $^\varrho\boldsymbol{f}$ Abtastwerte von Schriftzeichen, so ist $^\varrho\Omega$ die Bedeutung oder Klasse von $^\varrho\boldsymbol{f}$, die den zugehörigen Buchstaben angibt, $\boldsymbol{\Omega}$ ist eine Folge von Buchstaben oder ein Wort und $\widetilde{\boldsymbol{\Omega}}$ demnach die Menge der zulässigen Wörter. Ein Wörterbuch kann als zusätzliche oder auch als alleinige Maßnahme zur Kontextberücksichtigung dienen. Wenn man mit dem VITERBI-Algorithmus die Folge $\boldsymbol{\Omega}$ mit größter a posteriori Wahrscheinlichkeit bestimmt hat, wird $\boldsymbol{\Omega}$ nur dann akzeptiert, wenn es im Wörterbuch enthalten ist. Um

FOR $\varrho = 1$ TO N (also für alle Spalten)
FOR $\lambda = 1$ To k (also für alle Knoten einer Spalte)
berechne $G_{\lambda\varrho} = \ln p({}^{\varrho}\mathfrak{c} \mid \Omega_\lambda) + \max_{j\in\{1,\ldots,k\}}\{G_{j,\varrho-1} + \ln p_u(\Omega_\lambda \mid \Omega_j)\}$; setze für $\varrho = 1$ das $\max_j\{G_{j,\varrho-1} + \ln p_u(\Omega_\lambda \mid \Omega_j)\} = \ln p_u(\Omega_\lambda \mid \#)$ oder $\ln p(\Omega_\lambda)$; speichere die Indizes des Vorgängerknotens, der zum Maximum von $G_{\lambda\varrho}$ führt
in der letzten Spalte ($\varrho = N$) liegen k Gewichte $G_{\lambda N}$, $\lambda = 1, \ldots, k$ vor; man bestimme das größte dieser Gewichte $G_{\kappa N} = \max_\lambda G_{\lambda N}$; es gehört zu dem Pfad mit maximalem Gewicht
man ermittle den Pfad maximalen Gewichtes, indem man vom Knoten (κ, N) aus die Zeiger zurück zum Startknoten verfolgt; die Folge dieser Knoten gibt die Folge $\mathbf{\Omega}$;

Bild 4.7.3: Bestimmung des Pfades maximalen Gewichts

die Robustheit gegenüber einzelnen Fehlklassifikationen von Mustern ${}^{\varrho}\boldsymbol{f}$ zu erhöhen, werden oft die m Folgen $\mathbf{\Omega}_i\,, i = 1, \ldots, m$ mit größter a posteriori Wahrscheinlichkeit bestimmt. Dieses kann beispielsweise wie es oben als statistischer Ansatz beschrieben wurde geschehen oder indem man einfach alle kombinatorisch möglichen Folgen bildet, die sich aus einer bestimmten Anzahl alternativer Klassifikationen jedes Musters ${}^{\varrho}\boldsymbol{f}$ ergeben. Man vergleicht die Folgen $\mathbf{\Omega}_i$ der Reihe nach mit dem Wörterbuch. Ist die mit höchster a posteriori Wahrscheinlichkeit ein gültiges Wort, wird sie als das richtige Wort betrachtet und sonst die mit zweithöchster a posteriori Wahrscheinlichkeit untersucht, usw. Ein rascher Zugriff zum Wörterbuch ist über eine Hash–Codierung möglich.

n–Gramme

Statt vollständige Wörter der Länge N in einem Wörterbuch zu speichern, kann man auch zulässige Folgen von zwei, drei,... oder n Buchstaben (oder Klassen), die als n**-Gramme** bezeichnet werden, speichern und unter Umständen auch ihre Auftrittswahrscheinlichkeiten. Die Übergangswahrscheinlichkeiten $p_u(\Omega_\kappa\,|\Omega_\lambda)$ werden als *Bigramm*–Wahrscheinlichkeiten für das Paar $(\Omega_\lambda, \Omega_\kappa)$ bezeichnet. Eine weitere Spezialisierung besteht noch darin, diese n–Gramme abhängig von der Position im Wort zu ermitteln. Wegen Einzelheiten der Anwendung der n-Gramme zur Kontextberücksichtigung wird auf die Literatur verwiesen. Ein Vorteil der n–Gramme besteht darin, dass sie unabhängig vom Umfang des Lexikons sind, vorausgesetzt sie wurden mit einer repräsentativen Stichprobe bestimmt.

Relaxation

Kontext wird in *Relaxationsverfahren* dadurch berücksichtigt, dass man die Wahrscheinlichkeiten möglicher alternativer Bedeutungen eines Musters iterativ in Abhängigkeit von den Bedeutungen benachbarter Muster verändert. Um bei der bisherigen Notation zu bleiben, nehmen wir an, dass zum Muster ${}^{j}\boldsymbol{f}$ die möglichen Klassen oder Bedeutungen $\Omega^j \subset \{\Omega_1, \ldots, \Omega_k\}$ gehören. Diese können zum Beispiel diejenigen Klassen sein, deren a posteriori Wahrscheinlichkeit über einer vorgegebenen Schwelle liegt. Damit lässt sich jeder Klasse oder Bedeutung $\Omega_\kappa \in \Omega^j$ die Wahrscheinlichkeit $p(\Omega_\kappa\,|{}^{j}\boldsymbol{f})$ zuordnen, mit der Ω_κ bei Beobachtung von ${}^{j}\boldsymbol{f}$ auftritt. Für jedes Paar von Mustern ${}^{i}\boldsymbol{f}$, ${}^{j}\boldsymbol{f}$ wird es eine Menge $\Omega^{ij} \subseteq \Omega^i \times \Omega^j$ von kompatiblen Bedeutungen geben; eine Bedeutung $(\Omega_\kappa, \Omega_\lambda) \in \Omega^{ij}$ ist kompatibel mit ${}^{i}\boldsymbol{f}$, ${}^{j}\boldsymbol{f}$, wenn in der Nachbarschaft

eines Musters ${}^i\boldsymbol{f}$ mit der Bedeutung Ω_κ ein anderes Muster ${}^j\boldsymbol{f}$ die Bedeutung Ω_λ haben kann. Die oben erwähnten Bigramme geben solche kompatiblen Bedeutungen für zwei aufeinander folgende Zeichen an. Weiterhin wird angenommen, dass die Kompatibilität von Klasse Ω_κ für ${}^i\boldsymbol{f}$ mit der Klasse Ω_λ für ${}^j\boldsymbol{f}$ durch einen **Kompatibilitätskoeffizienten** $r_{ij}(\kappa,\lambda)$, der Werte zwischen -1 und $+1$ annimmt, bewertet werden kann. Dabei soll ein Wert nahe bei -1 anzeigen, dass die Klassen $(\Omega_\kappa,\Omega_\lambda)$ sehr selten für ${}^i\boldsymbol{f}$, ${}^j\boldsymbol{f}$ auftreten, ein Wert nahe bei $+1$ soll anzeigen, dass diese Klassen sehr häufig für ${}^i\boldsymbol{f}$, ${}^j\boldsymbol{f}$ auftreten, und ein Wert um 0 bedeutet, dass Ω_κ und Ω_λ relativ unabhängig sind.

Nach diesen anfänglichen Zuordnungen, die ähnlich auch beim statistischen Ansatz in (4.7.4) erforderlich sind, werden die Wahrscheinlichkeiten $p(\Omega_\kappa|\,{}^j\boldsymbol{f})$ iterativ verändert, wobei im Folgenden m den Iterationsschritt bezeichnet. Zunächst wird der Koeffizient

$$\beta_{im}(\Omega_\kappa)=\sum_j \alpha_{ij}\sum_\lambda r_{ij}(\kappa,\lambda)\,p_m(\Omega_\lambda|\,{}^j\boldsymbol{f})\;,\quad \sum_j\alpha_{ij}=1 \tag{4.7.8}$$

definiert, wobei die α_{ij} Gewichtsfaktoren sind. Dieser Koeffizient hat nur dann einen relativ großen positiven Wert, wenn es für ein Muster ${}^i\boldsymbol{f}$ mit der Bedeutung Ω_κ andere Muster ${}^j\boldsymbol{f}$ gibt, deren Bedeutung(en) Ω_λ eine hohe Wahrscheinlichkeit haben und stark kompatibel mit Ω_κ sind (d. h. $r_{ij}(\kappa,\lambda)\simeq 1$). Dieses wird als Indiz dafür genommen, dass Ω_κ zu ${}^i\boldsymbol{f}$ „passt“, und daher wird die Wahrscheinlichkeit $p_m(\Omega_\kappa|\,{}^i\boldsymbol{f})$ erhöht gemäß

$$p_{m+1}(\Omega_\kappa|\,{}^i\boldsymbol{f})=p_m(\Omega_\kappa|\,{}^i\boldsymbol{f})\frac{1+\beta_{im}(\Omega_\kappa)}{\sum_\kappa p_m(\Omega_\kappa|\,{}^i\boldsymbol{f})(1+\beta_{im}(\Omega_\kappa))}\;. \tag{4.7.9}$$

Stark negative Werte von $\beta_{im}(\Omega_\kappa)$ lassen sich analog interpretieren und führen zu einer Erniedrigung von $p_m(\Omega_\kappa|\,{}^i\boldsymbol{f})$ während Werte $\beta_{im}(\Omega_\kappa)\simeq 0$ keine Veränderung bewirken. Die Iteration wird einige Male für alle Klassen eines Musters und alle Muster ausgeführt. Der Idealfall ist der, dass von den möglichen Klassen $\Omega^j\subset\{\Omega_1,\ldots,\Omega_k\}$ des Musters ${}^j\boldsymbol{f}$ genau eine eine a posteriori Wahrscheinlichkeit nahe Eins erhält und dieses für alle Muster zutrifft. Erfahrungen mit Relaxationsverfahren bei verschiedenartigen Aufgaben haben gezeigt, dass meistens schon wenige Iterationen für stabile Ergebnisse genügen. Ein Beispiel für ihre Anwendung liegt bei der Nachverarbeitung von Bildpunkten eines Multispektralbildes, die zuvor unabhängig voneinander mit einem der üblichen Verfahren klassifiziert wurden.

4.8 Unüberwachtes Lernen (VA.1.2.3, 13.04.2004)

4.8.1 Anliegen

Die Ermittlung der Klassenbereiche, also der Teilbereiche des Merkmalsraums $\mathbb{R}_{\boldsymbol{c}}$, die den einzelnen Klassen zugeordnet sind, erfolgt durch Verarbeitung einer Stichprobe ω von Mustern. Dabei sind folgende zwei Fälle zu unterscheiden:

1. Von jedem Muster ${}^{\varrho}\boldsymbol{f} \in \omega$ ist die richtige Klasse *bekannt*, die Muster in der Stichprobe sind also klassifiziert, bzw. die Stichprobe ω ist zerlegt in k Teilmengen ω_κ gemäß (4.2.1), S. 323. Die Zusatzinformation in (1.3.1), S. 19, hat dann die Form $y_\varrho \in \{1, \ldots, \kappa, \ldots, k\}$. In diesem Fall zerfällt auch das Lernen in k unabhängige Einzelprobleme, je eines für jede der k Klassen. Es ist intuitiv klar und wird auch durch die entsprechenden theoretischen Ergebnisse bestätigt, dass dieses Problem relativ einfach ist. Man bezeichnet einen Lernprozess, der mit klassifizierten Mustern ausgeführt wird, auch als „überwachten“ Lernprozess („Lernen *mit* Lehrer“). Beim **überwachten Lernen** sind wiederum zwei Fälle zu unterscheiden:

   1.1 Die Bestimmung der Klassenbereiche erfolgt mit klassifizierten Mustern einmal vorweg (Lernphase), danach werden Muster ohne Veränderung des Klassifikators klassifiziert (Klassifikationsphase). Ein wiederholter Wechsel zwischen Lern- und Klassifikationsphase ist dem System nicht möglich. Beispiele dafür wurden in Abschnitt 4.2 – Abschnitt 4.6 behandelt.

   1.2 Die Klassenbereiche können mit klassifizierten Mustern laufend verändert werden, bzw. das System kann wiederholt zwischen Lern- und Klassifikationsphase wechseln. Die Basis dafür sind rekursive Parameterschätzungen, die in Abschnitt 4.2.3 und Abschnitt 4.4.3 kurz erwähnt wurden.

2. Die richtige Klasse eines Musters ${}^{\varrho}\boldsymbol{f} \in \omega$ ist *nicht bekannt*, die Muster in der Stichprobe sind also nicht klassifiziert, bzw. die Stichprobe ω ist nicht gemäß (4.2.1) in Teilmengen ω_κ zerlegt.

   Das Lernen muss hier im Prinzip gemeinsam für alle k Klassen durchgeführt werden. Dieses Problem ist im Vergleich zu Fall 1 wesentlich komplizierter. Man bezeichnet Lernen mit *nicht* klassifizierten Mustern als **unüberwachtes Lernen** („Lernen *ohne* Lehrer“).

   Auch hier sind zwei Fälle zu unterscheiden:

   2.1 Mit nicht klassifizierten Mustern sind die Klassenbereiche einmal vorweg zu bestimmen, danach bleibt der Klassifikator unverändert.

   Dieser Fall wird, wie in der Literatur üblich, als unüberwachtes Lernen bezeichnet; dazu werden auch Verfahren zur Analyse von Häufungsgebieten (*“cluster analysis”*) gerechnet. Beispiele dafür werden in Abschnitt 4.8.2 bis Abschnitt 4.8.5 behandelt.

   2.2 Die Klassenbereiche sollen mit nicht klassifizierten Mustern laufend verändert werden.

   Die zugehörigen Verfahren werden ebenfalls als unüberwachtes Lernen bezeichnet. Beispiele dafür enthalten die Schätzgleichungen in Abschnitt 4.8.2 und Abschnitt 4.8.3 in ihrer Anwendung auf jeweils ein Muster.

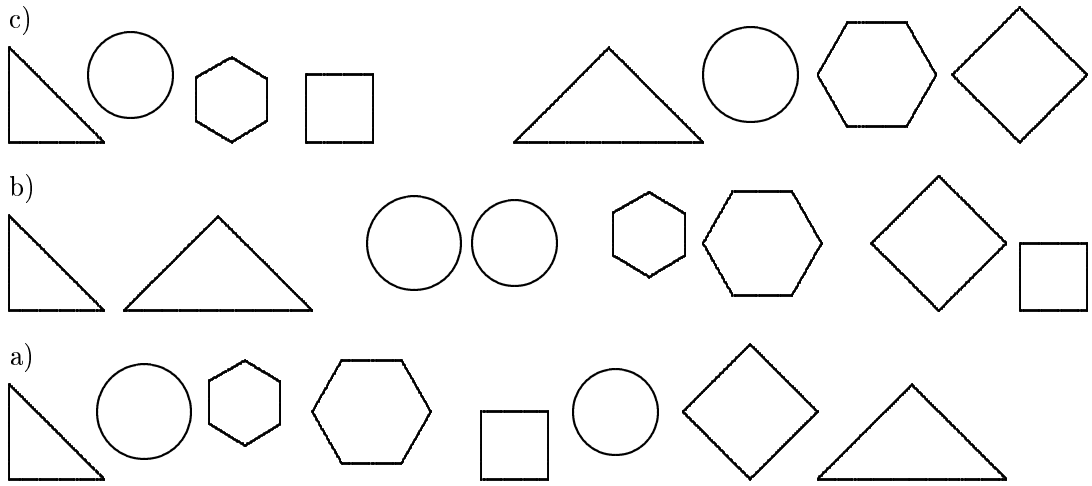


Bild 4.8.1: a) Eine Menge von Objekten ohne Klassenzugehörigkeit; b) Klassenbildung nach ähnlicher *Form*; c) Klassenbildung nach ähnlicher *Fläche*

3. Teilweise wird noch ein dritter Fall unterschieden, nämlich das Lernen mit einem *unvollkommenen* Lehrer. Damit wird der Fall bezeichnet, dass zwar Klassenzugehörigkeiten von Mustern in der Stichprobe gegeben sind, jedoch sind diese mit einer gewissen Wahrscheinlichkeit *falsch*. In der Regel wird hier wie beim überwachten Lernen verfahren und erwartet, dass sich die (hoffentlich wenigen) Fehler beim Training „herausmitteln“.

Zu den oben genannten Fällen 1.2, 2.1 und 2.2 liegen zwar zahlreiche Ergebnisse vor. Trotzdem ist der Fall 1.1 nach wie vor ein unverzichtbarer bzw. oft auch der einzige Schritt bei der Entwicklung von Klassifikationssystemen. Die erforderlichen klassifizierten Muster werden dabei über die Klassifikation einer Stichprobe durch den Entwickler gewonnen.

Sowohl theoretisch als auch praktisch ist die Frage äußerst interessant, wie man mit einer *nicht* klassifizierten Stichprobe Klassenbereiche ermitteln kann. Beispielsweise werden zur Entwicklung von Beleg– und Handschriftlesern Stichproben mit $10^4 - 10^6$ Mustern verarbeitet, sodass die Gewinnung einer klassifizierten Stichprobe erhebliche Arbeit verursacht. Von der Klassifikation von Elektrokardiogrammen (EKG) ist bekannt, dass es Grenzfälle gibt, in denen verschiedene Kardiologen das gleiche EKG unterschiedlich beurteilen; die „richtige“ Klasse ist also nicht immer zweifelsfrei festzustellen. Eine Haftpflichtversicherung, die bestimmte Daten ihrer Versicherten – wie Alter, Beruf, Wohnsitz – kennt und damit einige möglichst homogene Tarifklassen bilden möchte, muss die Zahl der Klassen oder den Begriff Homogenität festlegen, da hier, anders als bei den Beleglesern, Klassen nicht schon vorher gegeben sind.

Die grundsätzliche Problematik des unüberwachten Lernens verdeutlicht Bild 4.8.1. Je nach Entwurf des Algorithmus zum unüberwachten Lernen können sich die vier Klassen in Bild 4.8.1b) oder die zwei Klassen in Bild 4.8.1c) ergeben. Beide Klassenbildungen sind im Sinne des Algorithmus „richtig“, aber vielleicht nur eine davon wird die intuitive Vorstellung des menschlichen Betrachters oder die Erfordernisse eines Problemkreises treffen.

Zur Lösung des Problems des unüberwachten Lernens werden hier drei Vorgehensweisen vorgestellt.

- Die Verteilungsdichte der gesamten (unklassifizierten) Stichprobe wird durch eine Mi-

schungsverteilung repräsentiert, deren unbekannte Parameter geschätzt werden. Das Problem der Identifikation von Mischungsverteilungen wird in Abschnitt 4.8.2 behandelt.

- Es wird eine Gütefunktion vorgegeben, die den Wert (oder die Kosten) einer bestimmten Klasseneinteilung misst. Mit einem Optimierungsalgorithmus wird die Gütefunktion für die gegebene Stichprobe optimiert. Verfahren dafür werden in Abschnitt 4.8.4 betrachtet.
- Die Stichprobe wird als bewichteter Graph aufgefasst, der durch Eliminierung geeigneter Kanten in disjunkte Teilgraphen zerlegt wird, die den Klassen entsprechen. Dieses ist die Vorgehensweise in Abschnitt 4.8.5.
- Schließlich ist zu erwähnen, dass Verfahren wie die *Vektorquantisierung* (s. Abschnitt 2.1.4) und die *Merkmalskarte* (s. Abschnitt 4.5.4) ebenfalls unüberwacht eine Klasseneinteilung berechnen.

4.8.2 Die Identifikation von Mischungsverteilungen

Identifizierbarkeit

Wenn nur eine *unklassifizierte* Stichprobe ω gegeben ist und die Parameter eines statistischen Klassifikators bestimmt werden sollen, lassen sich die Schätzverfahren von Abschnitt 4.2.1 und Abschnitt 4.2.3 nicht anwenden. Der Grund ist, dass man nicht die bedingten Dichten $p(\boldsymbol{c}|\omega_\kappa)$ unabhängig voneinander schätzen kann, sondern nur die Mischungsverteilungsdichte für den Problemkreis Ω (in (4.2.15), S. 327, wurde bereits eine Mischungsverteilungsdichte je Klasse Ω_κ eingeführt)

$$p(\boldsymbol{c}) \;=\; \sum_{\kappa=1}^{k} p_\kappa p(\boldsymbol{c}|\omega_\kappa) = \sum_{\kappa=1}^{k} p_\kappa p(\boldsymbol{c}|\boldsymbol{a}_\kappa)\ ,$$

welche die unbekannten Parameter

$$\boldsymbol{\Theta} = \{k,\ \{p_\kappa, \boldsymbol{a}_\kappa | \kappa = 1, \ldots, k\}\} \tag{4.8.1}$$

enthält, sodass man zur Verdeutlichung auch $p(\boldsymbol{c}) = p(\boldsymbol{c}|\boldsymbol{\Theta})$ schreibt. Damit ein Klassifikator unüberwacht lernen kann, sind zwei Fragen zu klären:

1. Unter welchen Voraussetzungen lassen sich Schätzwerte $\widehat{\boldsymbol{\Theta}}$ für $\boldsymbol{\Theta}$ berechnen, d. h. welches sind die Bedingungen für die *Identifizierbarkeit* einer Mischungsverteilung?
2. Wie ist $\widehat{\boldsymbol{\Theta}}$ konkret zu berechnen, d. h. wie erfolgt die *Identifikation* der Parameter?

Diese Fragen sind geklärt, wie im Folgenden dargelegt wird. Dabei werden z. T. Verteilungsfunktionen $P(\boldsymbol{c})$, und nicht Dichten $p(\boldsymbol{c})$ betrachtet. Das Problem der Identifikation tritt sowohl im oben erwähnten Fall der unklassifiziertenStichprobe auf als auch in dem in Abschnitt 4.2.1 erwähnten Fall, dass man eine *klassifizierte* Stichprobe hat, aber *je Klasse* eine Mischungsverteilung ansetzt.

Entsprechend (4.1.1) wird vorausgesetzt, dass die bedingten Verteilungen $P(\boldsymbol{c}|\Omega_\kappa)$ der Merkmalsvektoren Elemente einer bekannten parametrischen Familie $\widetilde{P}(\boldsymbol{c}|\boldsymbol{a})$ sind, d. h. $P(\boldsymbol{c}|\Omega_\kappa) = P(\boldsymbol{c}|\boldsymbol{a}_\kappa)$ ist bis auf den Parametervektor $\boldsymbol{a}_\kappa$ bekannt. Es gebe eine mischende Verteilung,

$$P' = \{p_\kappa(\boldsymbol{a}_\kappa)|\kappa = 1, \ldots, k\}\ , \tag{4.8.2}$$

die $k < \infty$ Punkten $\boldsymbol{a}_\kappa$ eine Wahrscheinlichkeit $p_\kappa > 0$ zuordnet, wobei die Nebenbedingung

$$\sum_{\kappa=1}^{k} p_\kappa = 1 \ , \quad 1 \leq k < \infty \tag{4.8.3}$$

gilt. Durch die Abbildung

$$Q(P') = \sum_{\kappa=1}^{k} p_\kappa P(\boldsymbol{c}|\boldsymbol{a}_\kappa) = P(\boldsymbol{c}) \ , \quad P(\boldsymbol{c}|\boldsymbol{a}_\kappa) \in \widetilde{P}(\boldsymbol{c}|\boldsymbol{a}) \tag{4.8.4}$$

wird die n-dimensionale **Mischungsverteilung** $P(\boldsymbol{c})$ definiert. Ist $\widetilde{P}'$ die Menge der mischenden Verteilungen gemäß (4.8.2), (4.8.3), so ist

$$\widetilde{P}(\boldsymbol{c}) = Q(\widetilde{P}') = \{Q(P')|P' \in \widetilde{P}'\} \tag{4.8.5}$$

die Menge der (endlichen) Mischungsverteilungen.

Definition 4.14 *Unter* **Identifizierbarkeit** *der Menge der (endlichen) Mischungsverteilungen wird verstanden, dass sich für jedes $P(\boldsymbol{c}) \in \widetilde{P}(\boldsymbol{c})$ die Parameter $\boldsymbol{\Theta}$ in (4.8.1) eindeutig bestimmen lassen, d. h. dass*

$$P_1' \neq P_2' \Leftrightarrow Q(P_1') = P_1(\boldsymbol{c}) \neq P_2(\boldsymbol{c}) = Q(P_2') \ . \tag{4.8.6}$$

Die parametrische Familie $\widetilde{P}(\boldsymbol{c}|\boldsymbol{a})$ heißt identifizierbar, wenn die zugehörige Menge $\widetilde{P}(\boldsymbol{c})$ von Mischungsverteilungen identifizierbar ist.

Satz 4.19 *Eine notwendige und hinreichende Bedingung, dass die Menge $\widetilde{P}(\boldsymbol{c})$ der Mischungsverteilungen, die von der parametrischen Familie $\widetilde{P}(\boldsymbol{c}|\boldsymbol{a})$ erzeugt wird, identifizierbar ist, besteht darin, dass $\widetilde{P}(\boldsymbol{c}|\boldsymbol{a})$ eine linear unabhängige Menge von Funktionen ist.*
Beweis*: s. z. B. [Yakowitz und Spragins, 1968, Yakowitz, 1970]*

Die Bedingung ist notwendig, weil bei linear abhängigen Funktionen die gleiche Mischungsverteilung mit verschiedenen Parametern dargestellt werden könnte. Sie ist hinreichend, weil zwei verschiedene Darstellungen der gleichen Mischungsverteilung der Eigenschaft der eindeutigen Darstellung durch eine Basis widersprechen würden.

Eine eindeutige Schätzung der Parameter $\boldsymbol{\Theta}$ in (4.8.1) ist also nur möglich, wenn Satz 4.19 erfüllt ist. Es kann also prinzipiell *unlösbare* Probleme des unüberwachten Lernens geben, wenn die beschreibende Familie von Mischungsverteilungen *nicht* identifizierbar ist. Allerdings wird so ein Problem vermutlich *näherungsweise* lösbar sein, wenn man die fragliche Familie durch eine identifizierbare mit hinreichender Genauigkeit approximieren kann. Da man die „richtige“ Familie von Mischungsverteilungen einer konkreten Anwendung ohnehin nicht kennt, sondern dafür ein plausible Annahme macht, wird man natürlich zweckmäßigerweise eine identifizierbare Menge von Mischungsverteilungen zu Grunde legen. Unüberwachtes Lernen im Kontext statistischer Klassifikatoren ist also ein Problem der *Parameterschätzung* von identifizierbaren Mischungsverteilungen. Einige Ergebnisse zur Identifizierbarkeit sind in folgendem Satz zusammengefasst:

Satz 4.20 *Die Familie der n–dimensionalen Normalverteilungen (4.2.3), S. 324, ist identifizierbar, ebenso die der n–dimensionalen Exponentialverteilungen (4.2.11), S. 325.*
Ist $\widetilde{P}(c|\boldsymbol{a})$ eine eindimensionale identifizierbare Familie, so ist auch das Produkt aus n solcher Funktionen identifizierbar.
Beweis*: s. z. B. [Yakowitz, 1970]*

Maximum-likelihood-Schätzwert

Zur tatsächlichen Berechnung von Schätzwerten kommen wieder die Verfahren von Abschnitt 4.2.2 in Betracht, wobei hier als Beispiel die MLS gemäß (4.2.37), S. 333, betrachtet werden. Dabei wird die Klassenzahl k als *bekannt* vorausgesetzt.

Die Berechnung von MLS für unüberwachtes Lernen beruht auf (4.2.57). Da k hier als bekannt vorausgesetzt wurde, ist der Logarithmus der "likelihood"–Funktion

$$\begin{aligned} l(\{p_\kappa, \boldsymbol{a}_\kappa\}) &= \log\left[p(\omega\,|\,\{p_\kappa, \boldsymbol{a}_\kappa\})\right] \\ &= \sum_{j=1}^{N} \log\left[p({}^j\boldsymbol{c}\,|\,\{p_\kappa, \boldsymbol{a}_\kappa\})\right] \\ &= \sum_{j=1}^{N} \log\left[\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}\,|\,\boldsymbol{a}_\kappa)\right] \end{aligned} \tag{4.8.7}$$

bezüglich p_κ und $\boldsymbol{a}_\kappa$ zu maximieren mit der Nebenbedingung (4.8.3). Ohne eine konkrete Annahme über die Verteilungsdichten $p({}^j\boldsymbol{c}\,|\,\boldsymbol{a}_\kappa)$ erhält man folgendes Ergebnis:

Satz 4.21 *Bei bekannter Klassenzahl sind die MLS gegeben durch*

$$\widehat{p}_\lambda = \frac{1}{N}\sum_{j=1}^{N} \widehat{p}(\Omega_\lambda\,|\,{}^j\boldsymbol{c}) = \frac{1}{N}\sum_{j=1}^{N} \frac{\widehat{p}_\lambda p({}^j\boldsymbol{c}\,|\,\boldsymbol{a}_\lambda)}{\sum_{\kappa=1}^{k} \widehat{p}_\kappa p({}^j\boldsymbol{c}\,|\,\boldsymbol{a}_\kappa)}\,, \qquad \lambda = 1, \ldots, k \tag{4.8.8}$$

$$0 = \sum_{j=1}^{N} \widehat{p}(\Omega_\lambda\,|\,{}^j\boldsymbol{c}) \frac{\partial \log\left[p({}^j\boldsymbol{c}\,|\,\boldsymbol{a}_\lambda)\right]}{\partial a_{\lambda\mu}}\,, \quad \lambda = 1, \ldots, k\,, \quad \mu = 1, \ldots, m\,. \tag{4.8.9}$$

Zum *Beweis* von (4.8.8) wird mit einem LAGRANGE-Multiplikator ϑ die modifizierte likelihood–Funktion l' in (4.8.10) gebildet, die partielle Ableitung nach p_λ in (4.8.11) gebildet, das Ergebnis zu Null gesetzt, mit p_λ multipliziert und über λ summiert; daraus ergibt sich zunächst der Wert des LAGRANGE-Multiplikators in (4.8.14). Die modifizierte likelihood–Funktion l' ist

$$l'(\{p_\kappa, \boldsymbol{a}_\kappa\}) = \sum_{j=1}^{N} \log\left[\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}\,|\,\boldsymbol{a}_\kappa)\right] - \vartheta\left(\sum_{\kappa=1}^{k} p_\kappa - 1\right)\,. \tag{4.8.10}$$

Die partielle Ableitung nach p_λ ist

$$\begin{aligned} 0 &= \frac{\partial l'(\{p_\kappa, \boldsymbol{a}_\kappa\})}{\partial p_\lambda} \\ &= \frac{\partial}{\partial p_\lambda}\left\{\sum_{j=1}^{N} \log\left[\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}\,|\,\boldsymbol{a}_\kappa)\right] - \vartheta\left[\sum_{\kappa=1}^{k} p_\kappa - 1\right]\right\} \end{aligned}$$

$$= \sum_{j=1}^{N} \frac{p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\lambda)}{\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\kappa)} - \vartheta \; . \tag{4.8.11}$$

Multipliziert man mit p_λ und summiert über λ, ergibt sich

$$\begin{aligned} 0 &= \sum_{\lambda=1}^{k} p_\lambda \frac{\partial l'(\{p_\kappa, \boldsymbol{a}_\kappa\})}{\partial p_\lambda} \\ &= \sum_{\lambda=1}^{k} p_\lambda \left\{ \sum_{j=1}^{N} \frac{p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\lambda)}{\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\kappa)} - \vartheta \right\} \\ &= \sum_{j=1}^{N} \sum_{\lambda=1}^{k} \frac{p_\lambda p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\lambda)}{\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\kappa)} - \sum_{\lambda=1}^{k} p_\lambda \vartheta \; . \end{aligned} \tag{4.8.12}$$

Mit der bekannten Beziehung (4.1.34), S. 315, für die a posteriori Wahrscheinlichkeiten

$$p(\Omega_\lambda|\,{}^j\boldsymbol{c}) = \frac{p_\lambda p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\lambda)}{\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\kappa)} = \frac{p_\lambda p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\lambda)}{p({}^j\boldsymbol{c})} \tag{4.8.13}$$

erhält man schließlich

$$\begin{aligned} 0 &= \sum_{j=1}^{N} \sum_{\lambda=1}^{k} p(\Omega_\lambda|\,{}^j\boldsymbol{c}) - \vartheta \\ &= N - \vartheta \; , \\ \vartheta &= N \; . \end{aligned} \tag{4.8.14}$$

Damit ist zunächst der Wert des LAGRANGE-Multiplikators ϑ bestimmt. Mit den in (4.8.13) angegebenen a posteriori Wahrscheinlichkeiten $p(\Omega_\lambda|\,{}^j\boldsymbol{c})$ erhält man aus $p_\lambda \partial l'/\partial p_\lambda = 0$ die Beziehungen

$$\begin{aligned} 0 &= p_\lambda \frac{\partial l'(\{p_\kappa, \boldsymbol{a}_\kappa\})}{\partial p_\lambda} \\ &= p_\lambda \sum_{j=1}^{N} \frac{p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\lambda)}{\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\,\boldsymbol{a}_\kappa)} - \vartheta\, p_\lambda \\ &= \sum_{j=1}^{N} p(\Omega_\lambda|\,{}^j\boldsymbol{c}) - N p_\lambda \; , \\ \widehat{p}_\lambda &= \frac{1}{N} \sum_{j=1}^{N} \widehat{p}(\Omega_\lambda|\,{}^j\boldsymbol{c}) \; . \end{aligned} \tag{4.8.15}$$

Die letzte Gleichung ist gerade (4.8.8), sodass deren Beweis damit abgeschlossen ist. In (4.8.15) scheidet die Triviallösung $p_\lambda = 0$ aus, weil $p_\lambda > 0$ vorausgesetzt wurde.

Für die Parameter $\boldsymbol{a}_\lambda$ erhält man durch Ableiten und Nullsetzen

$$0 = \frac{\partial l'(\{p_\kappa, \boldsymbol{a}_\kappa\})}{\partial a_{\lambda\nu}}$$

$$
\begin{aligned}
&= \frac{\partial}{\partial a_{\lambda\nu}} \left\{ \sum_{j=1}^{N} \log \left[\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\boldsymbol{a}_\kappa) \right] - \vartheta \left(\sum_{\kappa=1}^{k} p_\kappa - 1 \right) \right\} \\
&= \sum_{j=1}^{N} \frac{p_\lambda}{\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\boldsymbol{a}_\kappa)} \cdot \frac{\partial p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)}{\partial a_{\lambda\nu}} \,. \qquad (4.8.16)
\end{aligned}
$$

Mit der Beziehung

$$
\begin{aligned}
\frac{p_\lambda}{\sum_{\kappa=1}^{k} p_\kappa p({}^j\boldsymbol{c}|\boldsymbol{a}_\kappa)} &= \frac{p_\lambda}{p({}^j\boldsymbol{c})} = \frac{p_\lambda}{p({}^j\boldsymbol{c})} \cdot \frac{p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)}{p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)} \\
&= \frac{p(\Omega_\lambda|{}^j\boldsymbol{c})}{p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)}
\end{aligned}
$$

ergibt sich für den MLS

$$
\begin{aligned}
0 &= \sum_{j=1}^{N} \frac{p(\Omega_\lambda|{}^j\boldsymbol{c})}{p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)} \cdot \frac{\partial p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)}{\partial a_{\lambda\nu}} \\
&= \sum_{j=1}^{N} p(\Omega_\lambda|{}^j\boldsymbol{c}) \, \frac{\partial \log\left[p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)\right]}{\partial a_{\lambda\nu}} \,,
\end{aligned}
$$

und das ist gerade (4.8.9), sodass auch dieser Teil von Satz 4.21 gezeigt ist.

Um zu einem konkreten Schätzwert für die Parameter $a_{\lambda\nu}$ zu kommen, ist eine Annahme über die bedingte Verteilungsdichte $p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)$ erforderlich; wie oben ausgeführt, kommt dafür nur eine *identifizierbare* Familie in Frage. Setzt man für $p({}^j\boldsymbol{c}|\boldsymbol{a}_\lambda)$ eine Normalverteilungsdichte mit den Parametern $\boldsymbol{\mu}_\lambda, \boldsymbol{\Sigma}_\lambda$, ergibt sich:

Satz 4.22 *Die MLS für Mittelwert und Kovarianzmatrix einer Mischung von Normalverteilungsdichten sind bei bekannter Klassenzahl*

$$
\begin{aligned}
\widehat{\boldsymbol{\mu}}_\lambda &= \frac{1}{N\widehat{p}_\lambda} \sum_{j=1}^{N} \widehat{p}(\Omega_\lambda|{}^j\boldsymbol{c}) \, {}^j\boldsymbol{c} = \frac{\sum_{j=1}^{N} \left(\widehat{p}(\Omega_\lambda|{}^j\boldsymbol{c}) \cdot {}^j\boldsymbol{c}\right)}{\sum_{j=1}^{N} \widehat{p}(\Omega_\lambda|{}^j\boldsymbol{c})} \,, \\
\widehat{\boldsymbol{\Sigma}}_\lambda &= \frac{1}{N\widehat{p}_\lambda} \sum_{j=1}^{N} \widehat{p}(\Omega_\lambda|{}^j\boldsymbol{c})({}^j\boldsymbol{c} - \widehat{\boldsymbol{\mu}}_\lambda)({}^j\boldsymbol{c} - \widehat{\boldsymbol{\mu}}_\lambda)^\mathsf{T} \,, \quad \lambda = 1, \ldots, k \,. \qquad (4.8.17)
\end{aligned}
$$

Beweis*: s. z. B. [Bock, 1974, Wolfe, 1967]*

Die obigen Schätzwerte lassen sich mit einem intuitiven Argument begründen. Angenommen man habe zwar keine Klasseneinteilung der Muster in der Stichprobe aber Werte für die Parameter. Zu jedem Muster lässt sich dann die a posteriori Wahrscheinlichkeit berechnen, die vermutlich für alle Klassen von Null verschiedene Werte aufweist. Beim unüberwachten Lernen gehört also jedes Muster aus der Stichprobe zu *jeder* Klasse mit einer Wahrscheinlichkeit zwischen Null und Eins, während beim überwachten Lernen jedes Muster mit der Wahrscheinlichkeit Eins zu *einer* Klasse gehört. Beim überwachten Lernen werden z. B. zur Berechnung des bedingten Mittelwertes $\boldsymbol{\mu}_\lambda$ alle Muster aufsummiert, die zur Stichprobe ω_λ gehören und

durch die Zahl N_λ der Muster aus ω_λ dividiert. Analog ist es plausibel, beim unüberwachten Lernen alle N Muster aus der Stichprobe ω zu addieren, da sie nun alle zur Klasse Ω_λ gehören, aber mit der Wahrscheinlichkeit $p(\Omega_\lambda\,|\,\boldsymbol{c})$ zu bewichten, da sie nur mit diesem Anteil zu Ω_λ gehören. Dividiert wird durch die Summe der Anteile, mit denen die Muster zu Ω_λ. Das ergibt gerade den Schätzwert in Satz 4.22; natürlich folgt aus dieser intuitiven Argumentation nicht, dass diese ein MLS ist.

Die Gleichungen (4.8.8), (4.8.17) enthalten die MLS für $p_\lambda, \boldsymbol{\mu}_\lambda, \boldsymbol{\Sigma}_\lambda$, $\lambda = 1, \ldots, k$, wenn normalverteilte Merkmale vorliegen. Man sieht, dass im Unterschied zu (4.2.8), (4.2.9), S. 325, die Schätzwerte *nicht* unabhängig voneinander berechnet werden können. Vielmehr ergibt sich durch die a posteriori Wahrscheinlichkeiten ein System *gekoppelter* transzendenter Gleichungen, dessen Lösung *nicht* in geschlossener Form möglich ist. Überwachtes und unüberwachtes Lernen unterscheiden sich also zwar nicht im Ansatz der Berechnung von Maximum-likelihood-Schätzwerten der unbekannten Parameter, jedoch ganz wesentlich in der Komplexität des resultierenden Schätzproblems.

Man sieht aber auch, dass die Gleichungen entkoppelt werden, wenn für eine Klasse die a posteriori Wahrscheinlichkeit Eins und für alle anderen Null wird. Die gekoppelten Gleichungen für unüberwachtes Lernen gehen dann in die entkoppelten für überwachtes Lernen über. Die Lösung des allgemeinen Gleichungssystems ist nur in Fällen mit wenigen Parametern möglich und damit praktisch wenig bedeutend; allerdings erlaubt die Nutzung des EM–Algorithmus die Schätzung der Parameter in praktisch interessanten Fällen. Dessen Grundidee, nämlich die Behandlung des Problems der fehlenden Information durch ein Iterationsverfahren, wurde bereits in Abschnitt 1.6.4 vorgestellt, einige Einzelheiten folgen im nächsten Abschnitt. Hier besteht die fehlende Information in der unbekannten Klassenzugehörgkeit eines Musters.

Die Berechnung von BAYES-Schätzwerten für unüberwachtes Lernen beruht auf (4.2.40), S. 334. Analog zu (4.2.63) ist die a posteriori Dichte für $\boldsymbol{\Theta}$

$$p(\boldsymbol{\Theta}\,|\,\omega) \;=\; \frac{p({}^N\boldsymbol{c}\,|\,\boldsymbol{\Theta})p(\boldsymbol{\Theta}\,|\,{}^1\boldsymbol{c},\ldots,{}^{N-1}\boldsymbol{c})}{\int\limits_{\mathbb{R}_{\boldsymbol{\Theta}}} p({}^N\boldsymbol{c}\,|\,\boldsymbol{\Theta})p(\boldsymbol{\Theta}\,|\,{}^1\boldsymbol{c},\ldots,{}^{N-1}\boldsymbol{c})\,\mathrm{d}\boldsymbol{\Theta}} \tag{4.8.18}$$

zu berechnen und der MAPS $\widehat{\boldsymbol{\Theta}}$ gemäß (4.2.40) zu bestimmen. Das Problem ist hier, dass es keine hinreichende Statistik für $\boldsymbol{\Theta}$ und keine selbstreproduzierenden Dichten $p(\boldsymbol{\Theta})$ gibt; die daraus resultierenden Schwierigkeiten wurden in Abschnitt 4.2.3 erörtert. Numerische Lösungen für $\widehat{\boldsymbol{\Theta}}$ wurden durch Diskretisierung des Parameter– und des Merkmalsraumes in einfachen Fällen berechnet.

4.8.3 Unüberwachte Berechnung von Schätzwerten

Die Ergebnisse des vorigen Abschnitts zum unüberwachten Lernen statistischer Klassifikatoren machen deutlich, dass vereinfachende näherungsweise Lösungen sowohl für die MLS in (4.8.8), (4.8.17) als auch die MAPS in (4.8.18) erforderlich sind. Der verbreitetste Ansatz dafür sind verschiedene Formen des **entscheidungsüberwachten Lernens**, dessen Systematisierung und theoretische Fundierung der EM–Algorithmus in Bild 1.6.1, S. 39, ist. Vereinfachend kann man sagen, dass der unüberwachte Lernprozess im Wesentlichen auf einen überwachten reduziert wird, indem man zuerst für ein neu beobachtetes Muster oder auch die gesamte Stichprobe ein Maß für die Klassenzugehörigkeit berechnet, dann dieses verwendet, um die Parameterschätzung zu verbessern, und diese Schritte bis zur Konvergenz iteriert; das ist auch das Prinzip

der Vorgehensweise in Abschnitt 4.8.4 in (4.8.34).

Berechnung von MLS mit dem EM–Algorithmus

Die Berechnung von MLS der Parameter einer Mischungsverteilung mit dem *EM–Algorithmus* besteht aus den Schritten

- initialisiere Startwerte der unbekannten Parameter;
- berechne Schätzwerte der a posteriori Wahrscheinlichkeiten – der E–Schritt (estimation);
- berechne damit neue Schätzwerte der Parameter – der M–Schritt (maximization)
- wiederhole den E– und M–Schritt bis zur Konvergenz.

Damit wird ein *lokales* Optimum erreicht.

Als Beispiel wird die Schätzung der Parameter einer Mischungsverteilung *je Klasse* aus einer klassifizierten Stichprobe betrachtet. Wie in (4.2.15), S. 327, ist die bedingte Verteilungsdichte von Mustern der Klasse Ω_κ gegeben durch

$$p(\boldsymbol{c}|\Omega_\kappa) = \sum_{l=1}^{L_\kappa} p_{\kappa,l}\, \mathcal{N}(\boldsymbol{c}|\boldsymbol{\mu}_{\kappa,l}, \boldsymbol{\Sigma}_{\kappa,l}) \;, \quad \sum_l p_{\kappa,l} = 1 \;. \tag{4.8.19}$$

Die Rechnung wird gestartet mit $L_\kappa = 1$ und Startwerten für die Parameter $\{p_\kappa,\, \boldsymbol{\mu}_\kappa,\, \boldsymbol{\Sigma}_\kappa\}$, die problemlos als MLS berechnet werden können. Die Iterationen des EM–Algorithmus beginnen mit $L_\kappa = 2$ und Startwerten für die zwei Mittelwerte, die analog zu (2.1.47), S. 75, bestimmt werden, und Startwerten für die anderen Parameter

$$\begin{aligned} \boldsymbol{\mu}_{\kappa,1} &= \boldsymbol{\mu}_\kappa + \delta \;, \quad \boldsymbol{\mu}_{\kappa,2} = \boldsymbol{\mu}_\kappa - \delta \;, \\ p_{\kappa,l} &= p_\kappa \;, \qquad \boldsymbol{\Sigma}_{\kappa,l} = \boldsymbol{\Sigma}_\kappa \;, \quad l = 1,2 \;. \end{aligned} \tag{4.8.20}$$

Diese Startwerte werden iterativ verbessert, indem zunächst gemäß (4.8.13) die a posteriori Wahrscheinlichkeiten für jede Mischungskomponente l der Klasse Ω_κ und jedes Muster ${}^\varrho\boldsymbol{c}$ berechnet werden zu

$$p(l|{}^\varrho\boldsymbol{c}, \kappa) = \frac{\widehat{p}_{\kappa,l}\, p({}^\varrho\boldsymbol{c}|\boldsymbol{\mu}_{\kappa,l}, \boldsymbol{\Sigma}_{\kappa,l})}{\sum_{l'} \widehat{p}_{\kappa,l'}\, p({}^\varrho\boldsymbol{c}|\boldsymbol{\mu}_{\kappa,l'}, \boldsymbol{\Sigma}_{\kappa,l'})} \;. \tag{4.8.21}$$

Mit diesen werden gemäß (4.8.8) und (4.8.17) neue Schätzwerte der unbekannten Parameter berechnet aus

$$\begin{aligned} \widehat{p}_{\kappa,l} &= \frac{1}{N_\kappa} \sum_{\varrho=1}^{N_\kappa} p(l|{}^\varrho\boldsymbol{c}, \kappa) \;, \\ \widehat{\boldsymbol{\mu}}_{\kappa,l} &= \sum_{\varrho=1}^{N_\kappa} \frac{p(l|{}^\varrho\boldsymbol{c}, \kappa)}{\sum_{\varrho'} p(l|{}^{\varrho'}\boldsymbol{c}, \kappa)}\; {}^\varrho\boldsymbol{c} \;, \\ \widehat{\boldsymbol{\Sigma}}_{\kappa,l} &= \sum_{\varrho=1}^{N_\kappa} \frac{p(l|{}^\varrho\boldsymbol{c}, \kappa)}{\sum_{\varrho'} p(l|{}^{\varrho'}\boldsymbol{c}, \kappa)}\; \left({}^\varrho\boldsymbol{c} - \widehat{\boldsymbol{\mu}}_{\kappa,l}\right)\left({}^\varrho\boldsymbol{c} - \widehat{\boldsymbol{\mu}}_{\kappa,l}\right)^\mathsf{T} \;, \quad l = 1, \ldots, L_\kappa \;. \end{aligned} \tag{4.8.22}$$

Die Rechnungen in (4.8.21) und (4.8.22) werden iteriert bis zur Konvergenz. Nun wird geprüft, welche Komponente der Mischung die Bedingung

$$\sum_{\varrho=1}^{N_\kappa} p(l|{}^\varrho\boldsymbol{c}, \boldsymbol{\mu}_{\kappa,l}, \boldsymbol{\Sigma}_{\kappa,l}) > \theta \tag{4.8.23}$$

erfüllt. Eine solche Komponente wird in zwei Mischungskomponenten zerlegt, deren Startparameter analog zu (4.8.20) bestimmt werden, es wird erneut die Iteration mit (4.8.21) und (4.8.22) bis zur Konvergenz ausgeführt, und es wird erneut (4.8.23) gepüft. Diese Schritte werden wiederholt, bis sich nichts mehr ändert.

Varianten dieser Vorgehensweise bestehen darin, die Kovarianzmatrizen $\mathbf{\Sigma}_{\kappa,l}$ auf *Diagonalmatrizen* einzuschränken; die Auffrischung der Parameter in (4.8.22) *nicht für alle* Mischungskomponenten durchzuführen, sondern nur für die eine mit maximaler a posteriori Wahrscheinlichkeit in (4.8.21); die Gleichungen (4.8.21) und (4.8.22) *je Muster* oder *je Stichprobe* zu durchlaufen; die Iteration nicht mit zwei Mischungskomponenten zu starten sondern mit einer Anzahl, die durch vorangehende Analyse von Häufungsgebieten bestimmt wurde, z. B. mit Hilfe der Vektorquantisierung in Abschnitt 2.1.4 oder mit Verfahren aus Abschnitt 4.8.4; oder die (beim EM–Algorithmus als bekannt vorausgesetzte aber unbekannte) Klassenanzahl nicht über (4.8.23) zu bestimmen, sondern durch Verwendung von Paaren von Mischungskomponenten, wofür auf die zitierte Literatur verwiesen wird.

Berechnung von MAPS

Zur Berechnung von MAPS (bzw. von BAYES-Schätzwerten) wird auf die *entkoppelten* rekursiven Gleichungen (4.2.73) zurückgegriffen, wobei die Entkopplung heuristisch durch eine Variante der Entscheidungsüberwachung erfolgt; dieses ist im Prinzip auch für die MLS nach (4.2.59) möglich sowie für die nichtrekursiven Versionen dieser Schätzwerte. Die Vorgehensweise ist ähnlich wie oben, wobei wir hier als Beispiel von einer unklassifizierten Stichprobe und nur einer Mischungskomponente je Klasse ausgehen.

Zunächst werden Startwerte der Parameter bestimmt, die entweder zufällig gewählt werden oder, natürlich wesentlich besser, aus einer kleinen klassifizierten Stichprobe berechnet werden. Mit diesen Startwerten werden für ein Muster aus der Stichprobe (für die rekursive Parameterschätzung) oder für alle Muster der Stichprobe (für die nichtrekursive Parameterschätzung) die a posteriori Wahrscheinlichkeiten der Klassen berechnet. Bei der harten oder festen Entscheidungsüberwachung wird das Muster mit (4.1.33), S. 314, klassifiziert und die zugeordnete Klasse als *richtige* Klasse betrachtet. Statt mit (4.8.18) werden danach die Parameter der ermittelten Klasse mit (4.2.73) verbessert. Mit den verbesserten Parametern wird wiederum neu klassifiziert und danach neue Parameter berechnet. Die Schritte Klassifikation und Parameterschätzung werden bis zur Konvergenz iteriert. Da eine nicht randomisierte *Entscheidung* vom System getroffen und der Lernprozess dadurch *überwacht* wird, bezeichnet man diese Vorgehensweise als (feste) *Entscheidungsüberwachung*. Die Konvergenz ist experimentell zu prüfen.

Statt der festen Entscheidungsüberwachung kann eine randomisierte verwendet werden. Mit den vorhandenen Schätzwerten für die Parameter werden Schätzwerte $\widehat{p}(\Omega_\lambda|{}^j\boldsymbol{c}),\ \lambda = 1,\ldots,k$ für die a posteriori Wahrscheinlichkeiten gemäß (4.1.34), S. 315, berechnet. Das beobachtete Muster ${}^j\boldsymbol{c}$ wird der Klasse Ω_κ mit der Wahrscheinlichkeit $\widehat{p}(\Omega_\kappa|{}^j\boldsymbol{c})$ zugeordnet. Die Parameter dieser Klasse werden mit (4.2.59) (MLS) oder (4.2.73) (MAPS) verbessert. Auch hier wird bis zur Konvergenz iteriert. Zu Aussagen über die Konvergenz, die unter bestimmten Bedingungen gesichert werden kann, wird auf die Literatur verwiesen.

In diesem Sinne ist der EM–Algorithmus die Variante der Entscheidungsüberwachung, bei der nicht nur einer, sondern alle Parameter aufgefrischt werden, und zwar mit einem Anteil, der ihrer a posteriori Wahrscheinlichkeit entspricht.

In jedem Falle ist eine wesentliche Verbesserung des Konvergenzverhaltens zu erwarten, wenn man nicht mit beliebigen Startwerten für die Parameter beginnt, sondern mit einigermaßen

zuverlässigen Schätzungen, die z. B. mit einer kleinen klassifizierten Stichprobe gemäß (4.2.8), (4.2.9) berechnet wurden.

4.8.4 Analyse von Häufungsgebieten

In diesem Abschnitt werden Methoden zum *unüberwachten Lernen* vorgestellt, bei denen die Ermittlung von Klassen in einer Stichprobe im Vordergrund steht, aber nicht das Training eines Klassifikators; dieses ließe sich mit der zerlegten Stichprobe nachträglich überwacht ausführen. Zu diesem Gebiet der Analyse von Häufungsgebieten ("cluster analysis") gibt es zahlreiche Ansätze, von denen hier einige als Beispiel erläutert werden. Die Grundlage aller Verfahren ist Postulat 6 von Abschnitt 1.3.

Abbildung in die Ebene

Einen ersten (subjektiven) Eindruck von der Struktur einer unklassifizierten Stichprobe erhält man, wenn man die M–dimensionalen Muster $\boldsymbol{f}$ oder die n–dimensionalen Merkmalsvektoren $\boldsymbol{c}$ in eine Ebene (auf ein Display) abbildet, d. h. auf $n' = 2$–dimensionale Vektoren reduziert. Ähnliche Muster, die vermutlich in eine Klasse gehören, sind in einer grafischen Darstellung als benachbarte Punkte erkennbar. Die Klasseneinteilung erfolgt also interaktiv. Die Abbildung muss die im $\mathbb{R}^n$ vorhandenen Abstände möglichst gut im $\mathbb{R}^2$ wiedergeben. Die lineare Abbildung (3.2.2), wobei die Matrix $\boldsymbol{\Phi}$ mit Satz 3.8, S. 224, und dem Kern $\boldsymbol{Q}^{(1)}$ in (3.8.8), S. 225, berechnet wird, ist eine einfache Methode. Bessere Ergebnisse werden in der Regel von strukturerhaltenden *nichtlinearen* Abbildungen erwartet, jedoch zeigt ein Vergleich an verschiedenen Beispielen, dass die einfache lineare Abbildung oft ausreicht bzw. zumindest ein guter Startpunkt ist.

Im Allgemeinen wird eine Stichprobe $\omega = \{{}^{\varrho}\boldsymbol{c},\ \varrho = 1,\ldots,N\}$ von Mustern ${}^{\varrho}\boldsymbol{c} \in \mathbb{R}^n$ in eine Stichprobe $\omega' = \{{}^{\varrho}\boldsymbol{c}',\ \varrho = 1,\ldots,N\}$ von Mustern ${}^{\varrho}\boldsymbol{c}' \in \mathbb{R}^{n'}$, $n' < n$ abgebildet. Die Abbildung soll so sein, dass möglichst alle Abstände von Mustern aus ω gleich denen von Mustern aus ω' sind. Da dieses für $n' < n$ i. Allg. nicht exakt möglich ist, wird iterativ eine Abbildung bestimmt, die die Unterschiede in den Abständen minimiert. Dazu werden Abstände d_{jk} bzw. d'_{jk} für zwei Muster ${}^{j}\boldsymbol{c}, {}^{k}\boldsymbol{c} \in \omega$ bzw. ${}^{j}\boldsymbol{c}', {}^{k}\boldsymbol{c}' \in \omega'$ sowie ein Fehler $\varepsilon(\omega,\omega')$ für den Unterschied der Abstände der Muster definiert. Beispiele sind

$$\boxed{\begin{aligned} d_{jk} &= ({}^{j}\boldsymbol{c} - {}^{k}\boldsymbol{c})^{\mathsf{T}}({}^{j}\boldsymbol{c} - {}^{k}\boldsymbol{c})\ , \quad d'_{jk} = ({}^{j}\boldsymbol{c}' - {}^{k}\boldsymbol{c}')^{\mathsf{T}}({}^{j}\boldsymbol{c}' - {}^{k}\boldsymbol{c}')\ , \\ \varepsilon(\omega,\omega') &= \frac{\sum\limits_{j=2}^{N}\sum\limits_{k=1}^{j-1} d_{jk}^{p}(d_{jk} - d'_{jk})^2}{\sum\limits_{j=2}^{N}\sum\limits_{k=1}^{j-1} d_{jk}^{p+2}}\ , \end{aligned}} \tag{4.8.24}$$

wobei die einfache Version mit $p = 0$ bereits sehr gute Ergebnisse bringt. Durch ein Iterationsverfahren werden die Muster ${}^{\varrho}\boldsymbol{c}' \in \mathbb{R}^{n'}$ so verschoben, dass $\varepsilon(\omega,\omega')$ minimiert ist. Dieses ist offenbar für beliebige Werte von $n' < n$ möglich. Für Einzelheiten zu geeigneten Iterationsverfahren wird auf die Literatur verwiesen.

	$S(N,k)$			$B(N)$
N	$k=2$	$k=4$	$k=10$	
4	7	1	0	15
10	511	34 105	1	$1,16 \cdot 10^5$
20	$5,24 \cdot 10^5$	$4,52 \cdot 10^{10}$	$5,92 \cdot 10^{12}$	$5,17 \cdot 10^{13}$
50	$5,63 \cdot 10^{14}$	$5,29 \cdot 10^{28}$	$2,62 \cdot 10^{43}$	$1,86 \cdot 10^{47}$

Tabelle 4.1: Einige Beispiele für STERLING-Zahlen und BELL-Zahlen

Minimierung einer Kostenfunktion

Das Prinzip vieler Verfahren zur Ermittlung von Häufungsgebieten lässt sich kurz damit zusammenfassen, dass eine *Kostenfunktion* vorgegeben wird, welche die Kosten einer bestimmten Klasseneinteilung der Muster einer Stichprobe bewertet, sowie ein *Optimierungsalgorithmus* angegeben wird, der die Muster so auf Klassen verteilt, dass die Kosten minimiert werden.

Bei der Wahl der Kostenfunktion ergibt sich analog die in Bild 4.8.1 angedeutete Problematik, dass man mehr oder weniger intuitiv die Vorstellungen eines menschlichen Betrachters bzw. die Erfordernisse einer Anwendung in den gewählten Kosten erfassen muss. Dabei gibt es Kostenfunktionen, die nur Parameter bis zur Ordnung zwei enthalten, also Mittelwert und Kovarianzmatrix wie in (4.8.26), und solche, die auf im Prinzip beliebigen Verteilungsdichten basieren, wie z. B. Mischungen von Normalverteilungen (s. (4.2.15), S. 327) oder der (nichtparametrischen) PARZEN-Schätzung (s. (4.2.142), S. 354); ein Beispiel dafür ist das informationstheoretisch basierte Maß in (4.8.44). Solche Kostenfunktionen können auch als *Abstände von Verteilungsdichten* aufgefasst werden, wofür Beispiele bereits in (3.9.14) – (3.9.22), S. 251, gebracht wurden.

Ähnlich wie beim Problem der Merkmalsbewertung und –auswahl in Abschnitt 3.9 ist es auch hier aus Komplexitätsgründen nicht möglich, *alle* möglichen Zuordnungen von Mustern in Klassen systematisch auszuprobieren, um die im Sinne der Kostenfuntkion optimale zu finden. Man ist daher auf heuristische suboptimale Verfahren angewiesen, für die Beispiele in Bild 4.8.2 und Bild 4.8.3 angegeben werden. Dazu kommt das in (4.8.50) angegebene Beispiel für hierarchische Zerlegungen.

Die Zahl der Zerlegungen (Partitionen) einer Stichprobe vom Umfang N in genau k Klassen ist gleich der STERLING-Zahl zweiter Art $S(N,k)$. Die Zahl der Zerlegungen einer Stichprobe in $1, 2, \ldots,$ oder N Klassen ist gleich der BELL-Zahl $B(N)$. Es gilt

$$\begin{aligned} S(N,k) &= 0\,, \quad \text{für } k > N\,, \\ S(N,1) &= S(N,N) = 1\,, \\ S(N+1,k) &= S(N,k-1) + kS(N,k)\,, \\ B(N) &= \sum_{\kappa=1}^{N} S(N,\kappa)\,. \end{aligned} \tag{4.8.25}$$

Tabelle 4.1 gibt einige Beispiele, die zeigen, dass schon bei kleinem Stichprobenumfang die Zahl der Zerlegungen so groß wird, dass ein systematisches Ausprobieren aller Möglichkeiten zur Optimierung einer Kostenfunktion keine praktikable Lösung ist.

Minimierung einer parametrischen Kostenfunktion

Mit $\Phi(\boldsymbol{c}, \boldsymbol{a}_\kappa)$ werden die „Kosten“ bezeichnet, die sich bei Einordnung des Merkmalsvektors $\boldsymbol{c}$ in die Klasse Ω_κ ergeben, wobei die Information über Ω_κ im Parametervektor $\boldsymbol{a}_\kappa$ enthalten sei. Beispiele für Kostenfunktionen, die auf zweiten Momenten basieren, sind

$$\boxed{\begin{array}{lcl} \Phi_1(\boldsymbol{c}, \boldsymbol{a}_\kappa) & = & (\boldsymbol{c} - \boldsymbol{\mu}_\kappa)^2 \;, \quad \boldsymbol{a}_\kappa = \boldsymbol{\mu}_\kappa \;, \\ \Phi_2(\boldsymbol{c}, \boldsymbol{a}_\kappa) & = & (\boldsymbol{c} - \boldsymbol{\mu}_\kappa)^\mathsf{T} \boldsymbol{\Sigma}_\kappa^{-1} (\boldsymbol{c} - \boldsymbol{\mu}_\kappa) \;, \quad \boldsymbol{a}_\kappa = (\boldsymbol{\mu}_\kappa, \boldsymbol{\Sigma}_\kappa) \;, \\ \Phi_3(\boldsymbol{c}, \boldsymbol{a}_\kappa) & = & |\boldsymbol{\Sigma}_\kappa|^{(1/n)} (\boldsymbol{c} - \boldsymbol{\mu}_\kappa)^\mathsf{T} \boldsymbol{\Sigma}_\kappa^{-1} (\boldsymbol{c} - \boldsymbol{\mu}_\kappa) \;. \end{array}} \tag{4.8.26}$$

In der ersten Gleichung lassen sich die Parameter $\boldsymbol{a}_\kappa = \boldsymbol{\mu}_\kappa$ als Klassenzentren oder **Prototypen** auffassen. Die mittleren Kosten bei der Klassifikation sind

$$V = \sum_{\kappa=1}^{k} p_\kappa \int\limits_{\Omega_\kappa} \Phi(\boldsymbol{c}, \boldsymbol{a}_\kappa)\, p(\boldsymbol{c}|\Omega_\kappa)\, \mathrm{d}\boldsymbol{c} \;. \tag{4.8.27}$$

Mit der Mischungsverteilungsdichte

$$p(\boldsymbol{c}) \;=\; \sum_{\kappa=1}^{k} p_\kappa p(\boldsymbol{c}|\Omega_\kappa) \tag{4.8.28}$$

und der Voraussetzung, dass sich die bedingten Dichten $p(\boldsymbol{c}|\Omega_\kappa)$ nicht überlappen, gilt auch

$$V = \sum_{\kappa=1}^{k} \int\limits_{\Omega_\kappa} \Phi(\boldsymbol{c}, \boldsymbol{a}_\kappa)\, p(\boldsymbol{c})\, \mathrm{d}\boldsymbol{c} \;. \tag{4.8.29}$$

Gesucht sind Parameter $\boldsymbol{a}_\kappa^*$ und Klassenbereiche Ω_κ^*, sodass die mittleren Kosten V in (4.8.29) minimiert werden, also

$$\{\boldsymbol{a}_\kappa^*, \Omega_\kappa^*\} = \underset{\boldsymbol{a}_\kappa, \Omega_\kappa}{\operatorname{argmin}}\, V(\boldsymbol{a}_\kappa, \Omega_\kappa) \;. \tag{4.8.30}$$

Die Zahl k der Klassen wird als bekannt vorausgesetzt. In (4.8.30) sind also sowohl die Klassenbereiche als auch die Parameter zu verändern. Mit der charakteristischen Funktion

$$\delta(\boldsymbol{c}, \boldsymbol{a}_\kappa) = \begin{cases} 1 & : \quad \boldsymbol{c} \in \Omega_\kappa \\ 0 & : \quad \text{sonst} \end{cases} \tag{4.8.31}$$

ergibt sich

$$V = \int\limits_{\mathbb{R}_{\boldsymbol{c}}} \sum_{\kappa=1}^{k} \delta(\boldsymbol{c}, \boldsymbol{a}_\kappa)\, \Phi(\boldsymbol{c}, \boldsymbol{a}_\kappa)\, p(\boldsymbol{c})\, \mathrm{d}\boldsymbol{c} \;. \tag{4.8.32}$$

Ähnlich wie in Abschnitt 4.1.3 wird V minimiert, wenn man jeden Merkmalsvektor $\boldsymbol{c}$ der Klasse mit minimalem $\Phi(\boldsymbol{c}, \boldsymbol{a}_\kappa)$ zuordnet, also die charakteristische Funktion durch

$$\delta(\boldsymbol{c}, \boldsymbol{a}_\kappa) = 1 \;, \quad \text{wenn } \Phi(\boldsymbol{c}, \boldsymbol{a}_\kappa) = \min_{\lambda} \Phi(\boldsymbol{c}, \boldsymbol{a}_\lambda) \tag{4.8.33}$$

wähle Kostenfunktion Φ, anfängliche Zahl der Klassen k, Startwerte $\boldsymbol{a}_{\kappa 0}, \kappa = 1, \ldots, k$ der Parameter; setze Iterationsindex $N = 0$
klassifiziere die Stichprobe
berechne neue Parameter $\boldsymbol{a}_{\kappa,N+1}$
wenn eine der Klassen zu inhomogen ist, *zerlege* sie in zwei neue
wenn eine der Klassen zu wenig Muster enthält, *verschmelze* sie mit einer anderen
wenn zwei Klassen zu dicht benachbart sind, *vereinige* sie zu einer
UNTIL die Parameter werden nicht mehr verändert, d. h. $\boldsymbol{a}_{\kappa,N+1} \approx \boldsymbol{a}_{\kappa,N}$

Bild 4.8.2: Schema der iterativen Bestimmung von Klassenbereichen

definiert. Da V ein Erwartungswert ist, liegt es nahe (4.8.33) und (1.6.30), S. 40, in der Iterationsvorschrift

$$\boldsymbol{a}_{\kappa,N+1} = \begin{cases} \boldsymbol{a}_{\kappa N} - \beta_N \nabla_{\boldsymbol{a}_\kappa} \Phi({}^N\boldsymbol{c}, \boldsymbol{a}_{\kappa N}) & : \quad \Phi({}^N\boldsymbol{c}, \boldsymbol{a}_{\kappa N}) = \min_\lambda \Phi({}^N\boldsymbol{c}, \boldsymbol{a}_{\lambda N}) \\ a_{\lambda N} & : \quad \forall\, \lambda \neq \kappa \end{cases} \tag{4.8.34}$$

zu kombinieren. Das läuft darauf hinaus, dass man zunächst für feste Parameterwerte die beste Klassenzuordnung sucht – hier also ein neues Muster gemäß (4.8.33) klassifiziert – und dann für feste Klassenzuordnung die besten Parameter bestimmt – hier also die Parameter iterativ verbessert. Beispielsweise erhält man für die Funktion Φ_1 in (4.8.26)

$$\boldsymbol{a}_{\kappa,N+1} = \begin{cases} \boldsymbol{a}_{\kappa N} + \beta_N({}^N\boldsymbol{c} - \boldsymbol{a}_{\kappa N}) & : \quad ({}^N\boldsymbol{c} - \boldsymbol{a}_{\kappa N})^2 = \min_\lambda ({}^N\boldsymbol{c} - \boldsymbol{a}_{\lambda N})^2 \\ \boldsymbol{a}_{\lambda N} & : \quad \forall\, \lambda \neq \kappa \end{cases} . \tag{4.8.35}$$

Abgesehen von unterschiedlichen Kostenfunktionen Φ sind weitere Modifikationen der obigen Vorgehensweise denkbar. Wenn eine Stichprobe ω mit N Mustern gegeben ist, kann man zunächst die Stichprobe mit (4.8.33) klassifizieren und dann die Parameter $\boldsymbol{a}_\kappa$ in (4.8.26) neu berechnen; die Schritte Klassifikation und Parameterberechnung werden wiederholt bis Klassenbereiche und Parameter konstant bleiben. Außerdem ist es möglich, die Zahl k der Klassen durch heuristische Kriterien zu beeinflussen. Beispiele für derartige Algorithmen sind ISODATA (*Iterative Self-Organizing Data Analysis Technique A*), Vektorquantisierung (s. Abschnitt 2.1.4) und andere. Ihr Prinzip zeigt Bild 4.8.2.

Das von einem Algorithmus dieses Typs gelieferte Ergebnis hängt von der gewählten Kostenfunktion Φ, dem Kriterium für Inhomogenität, der Mindestzahl der Muster je Klasse und dem Mindestabstand zweier Klassen ab. Ein Kriterium für Inhomogenität ist die Bimodalität einer der n bedingten eindimensionalen marginalen Verteilungsdichten der Merkmalsvektoren einer Klasse; die Mindestanzahl der Muster je Klasse wird als Bruchteil des Verhältnisses der Gesamtzahl der Muster in der Stichprobe zur aktuellen Klassenzahl gewählt; der Mindestabstand zweier Klassen kann als Bruchteil des mittleren Abstandes von Mustern in der gesamten Stichprobe gewählt werden. Eine Verallgemeinerung der charakteristischen Funktion (4.8.31) erhält man, wenn δ Werte zwischen Null und Eins annehmen darf. Das bedeutet, dass man $\boldsymbol{c}$ nicht genau einer Klasse zuordnet, sondern mit der durch δ gegebenen Sicherheit mehreren Klassen. Die darauf beruhenden Algorithmen werden als "fuzzy" ISODATA Algorithmen bezeichnet, und es lässt sich zeigen, dass diese unter bestimmten Voraussetzungen konvergieren.

Ein klassischer Vertreter für diesen Typ von Algorithmen ist der **k–means Algorithmus**, der daher kurz vorgestellt wird. Er bestimmt k Klassenbereiche in *einem* Durchlauf durch eine

Stichprobe ω mit N Mustern wie in (1.3.1), S. 19, jedoch ohne Zusatzinformation. Die ersten k Klassenzentren (oder Prototypen) sind die ersten k Muster aus der Stichprobe. In einer Verarbeitungsschleife werden dann für alle $N-k$ noch nicht betrachteten Muster die Schritte Klassifikation und Aktualisierung durchgeführt. Im Klassifikationsschritt wird das aktuell verwendete Muster aus der Stichprobe nach dem nächsten Klassenzentrum klassifiziert. Im Aktualisierungsschritt wird das Klassenzentrum, dem das aktuell verwendete Muster zugeordnet wurde, neu berechnet. Die damit erzielbaren Ergebnisse sind relativ zur Einfachheit des Algorithmus erstaunlich gut.

Minimierung einer nichtparametrischen Kostenfunktion

Ein nichtparametrisches Abstandsmaß lässt sich aus der RENYI-Entropie

$$H = \frac{1}{1-\alpha} \ln \left[\int p(\boldsymbol{c})^{\alpha} \, \mathrm{d}\boldsymbol{c} \right] \quad , \quad \alpha > 0 \; , \; \alpha \neq 1 \tag{4.8.36}$$

ableiten. Für die Verteilungsdichte $p(\boldsymbol{c})$ wird ein (nichtparametrischer) PARZEN-Schätzwert mit einer GAUSS-Fensterfunktion, in der $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$ gesetzt wird, verwendet

$$p(\boldsymbol{c}) = \frac{1}{N} \sum_{j=1}^{N} g_0 \left((\boldsymbol{c} - {}^j\boldsymbol{c}) \,|\, \sigma^2 \right) = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp \left[-\frac{1}{2} \left(\frac{(\boldsymbol{c} - {}^j\boldsymbol{c})}{\sigma} \right)^2 \right] \; . \tag{4.8.37}$$

Mit der Wahl $\alpha = 2$ und der Substitution von (4.8.37) in (4.8.36) ergibt sich

$$\begin{aligned} H &= -\ln \left[\int \left(\frac{1}{N} \sum_{j=1}^{N} \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp \left[-\frac{(\boldsymbol{c} - {}^j\boldsymbol{c})^2}{2\sigma^2} \right] \right) \right. \\ &\qquad \left. \times \left(\frac{1}{N} \sum_{k=1}^{N} \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp \left[-\frac{(\boldsymbol{c} - {}^k\boldsymbol{c})^2}{2\sigma^2} \right] \right) \, \mathrm{d}\boldsymbol{c} \right] \\ &= -\ln \left[\frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} \frac{1}{\sqrt{(4\pi\sigma^2)^n}} \exp \left[-\frac{({}^j\boldsymbol{c} - {}^k\boldsymbol{c})^2}{4\sigma^2} \right] \right] \\ &= -\ln \left[\frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} g_0 \left(({}^j\boldsymbol{c} - {}^k\boldsymbol{c}) \,|\, 2\sigma^2 \right) \right] \; . \end{aligned} \tag{4.8.38}$$

Damit liegt ein Entropiemaß vor, das *nichtparametrisch* ist, d. h. für (im Sinne der PARZEN-Schätzung) beliebige Verteilungsdichten formuliert ist, das *keine* Integration mehr erfordert, das mit der Komplexität $\mathcal{O}(N^2)$ berechenbar ist, wobei N wie üblich der Stichprobenumfang ist, und das nur noch einen freien Parameter, nämlich σ, enthält. Die darin auftretende Doppelsumme

$$V = \sum_{j=1}^{N} \sum_{k=1}^{N} g_0 \left(({}^j\boldsymbol{c} - {}^k\boldsymbol{c}) \,|\, 2\sigma^2 \right) \tag{4.8.39}$$

wird als *Informationspotential* bezeichnet; es nimmt mit wachsendem Abstand der Stichprobenelemente ab.

Wenn eine Stichprobe ω in zwei Teilmengen ω_κ bzw. ω_λ mit N_κ bzw. N_λ Elementen $\boldsymbol{c}_\kappa \in \omega_\kappa$ bzw. $\boldsymbol{c}_\lambda \in \omega_\lambda$ zerlegt wurde, wird das Informationspotential als Ansatz zur Definition einer Kostenfuntkion

$$\Phi(\omega_\kappa, \omega_\lambda) = \frac{1}{N_\kappa N_\lambda} \sum_{j=1}^{N_\kappa} \sum_{k=1}^{N_\lambda} g_0 \left(({}^j\boldsymbol{c}_\kappa - {}^k\boldsymbol{c}_\lambda) | 2\sigma^2 \right) \tag{4.8.40}$$

für diese beiden Teilmengen verwendet. Es wird eine Indikatorfunktion

$$\chi({}^j\boldsymbol{c}, {}^k\boldsymbol{c}) = \begin{cases} 0 & : \ {}^j\boldsymbol{c} \in \omega_\lambda \ \wedge \ {}^k\boldsymbol{c} \in \omega_\lambda \ , \quad \lambda \in \{1, \ldots, k\} \\ 1 & : \ \text{sonst} \end{cases} \tag{4.8.41}$$

definiert, die nur dann den Wert Eins annimmt, wenn zwei Merkmale $({}^j\boldsymbol{c}, {}^k\boldsymbol{c})$ zu *verschiedenen* Klassen gehören. Damit kann man auch schreiben

$$\Phi(\omega_\kappa, \omega_\lambda) = \frac{1}{N_\kappa N_\lambda} \sum_{j=1}^{N} \sum_{k=1}^{N} \chi({}^j\boldsymbol{c}, {}^k\boldsymbol{c}) g_0 \left(({}^j\boldsymbol{c} - {}^k\boldsymbol{c}) | 2\sigma^2 \right) \ . \tag{4.8.42}$$

In der obigen Doppelsumme werden nun durch die Indikatorfunktion nur solche Paare von Merkmalen ausgewählt, die zu verschiedenen Klassen gehören, wie es in (4.8.40) durch die zusätzliche Indizierung der Merkmalsvektoren mit dem Klassenindex auch erreicht wird. Diese zunächst für zwei Klassen definierte Kostenfunktion wird auf k Klassen verallgemeinert, indem man über alle verschiedenen Paare von Klassen mittelt

$$\boxed{\begin{aligned} \Phi_4(\omega_1, \ldots, \omega_k) &= \frac{2}{k(k-1)} \sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-1} \Phi(\omega_\kappa, \omega_\lambda) \\ &= \frac{2}{k(k-1)} \sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-1} \frac{1}{N_\kappa N_\lambda} \sum_{j=1}^{N_\kappa} \sum_{k=1}^{N_\lambda} g_0 \left(({}^j\boldsymbol{c}_\kappa - {}^k\boldsymbol{c}_\lambda) | 2\sigma^2 \right) \ . \end{aligned}} \tag{4.8.43}$$

Mit (4.8.41) lässt sich in Analogie zu (4.8.42) auch eine Kostenfunktion

$$\boxed{\Phi_5(\omega_1, \ldots, \omega_k) = \frac{1}{2N_1 N_2 \ldots N_k} \sum_{j=1}^{N} \sum_{k=1}^{N} \chi({}^j\boldsymbol{c}, {}^k\boldsymbol{c}) g_0 \left(({}^j\boldsymbol{c} - {}^k\boldsymbol{c}) | 2\sigma^2 \right)} \tag{4.8.44}$$

definieren. In einer guten Zerlegung der Stichprobe ist Φ_4 bzw. Φ_5 *klein*, da g_0 mit wachsendem Abstand von Merkmalsvektoren rasch abfällt und in verschiedenen Teilmengen nur relativ weit voneinander entfernte Merkmalsvektoren liegen sollten.

Die Aufgabe besteht nun darin, eine Zerlegung von ω zu finden, die Φ_4 bzw. Φ_5 *minimiert*. Dieses erfolgt in zwei Schritten, nämlich der Berechnung einer *anfänglichen Zerlegung* von ω und der anschließenden *Optimierung* von $\Phi(\omega)$.

Die *anfängliche Zerlegung* kann auf unterschiedliche Arten erfolgen, z. B. auch mit dem Algorithmus in Bild 4.8.2. Da noch ein Optimierungsschritt folgt, wird jedoch ein vereinfachter Algorithmus zur anfänglichen Zerlegung benutzt. Nach Vorgabe der Zahl k von Klassen werden Untermengen der Stichprobe vom Umfang $N_\mathrm{a} = N/k$ gebildet. Es werden N_a Elemente aus ω zufällig ausgewählt. Diese sind die Startpunkte für die Auswahl weiterer Elemente. Sei $\boldsymbol{c}_{\mathrm{a},1}$ der erste zufällig gewählte Merkmalsvektor, der das erste Element in der Untermenge $\omega_{\mathrm{a},1}$ wird.

initialisiere: Stichprobe ω, Optimierungsfunktion Φ, Klassenzahl k		
berechne anfängliche Zerlegung in Teilmengen $\omega_1, \ldots, \omega_k$ und Wert von Φ		
FOR alle Muster ${}^{\varrho}\boldsymbol{c} \in \omega,\ \varrho = 1, \ldots, N$		
	entfere das Muster ${}^{\varrho}\boldsymbol{c}$ aus der aktuell zugeordneten Klasse; diese sei ω_κ	
	FOR $\lambda = 1, \ldots, k,\ \lambda \neq \kappa$	
		bringe ${}^{\varrho}\boldsymbol{c}$ nach ω_λ, berechne neuen Wert Φ'_λ der Optimierungsfunktion
	berechne $\Phi'_{\min} = \min_{\lambda \neq \kappa} \Phi'_\lambda$; der zugehörige Wert von λ sei $\lambda_{\min}$	
	IF	$\Phi'_{\min} < \Phi$
	THEN	${}^{\varrho}\boldsymbol{c}$ wird aus ω_κ entfernt und nach $\omega_{\lambda_{\min}}$ gebracht; setze $\Phi = \Phi'_{\min}$
	ELSE	${}^{\varrho}\boldsymbol{c}$ verbleibt in ω_κ
UNTIL die Zerlegung wird nicht mehr geändert		

Bild 4.8.3: Zur Optimierung einer Klassenzerlegung durch Neuzuordnung von Mustern

Wenn bereits $N' < N_\mathrm{a}$ Elemente in $\omega_{\mathrm{a},1}$ sind, wird als $(N'+1)$–tes dasjenige aus ω bestimmt, dass am nächsten zu einem der N' vorhandenen liegt. Es werden also *nicht* die $N_\mathrm{a} - 1$ nächsten Nachbarn von $\boldsymbol{c}_{\mathrm{a},1}$ bestimmt. Dieser Prozess wird so oft wiederholt, bis N_a Elemente in $\omega_{\mathrm{a},1}$ sind, und dann für die restlichen Untermengen $\omega_{\mathrm{a},\lambda},\ \lambda = 2, k$ durchgeführt.

Die anschließende Optimierung wird so durchgeführt, dass ein Muster aus der Stichprobe ω, das z. B. aktuell in der Teilmenge ω_κ ist, aus seiner Teilmenge entfernt und versuchsweise in eine andere Teilmenge ω_λ gebracht wird. Wenn diese Neuzuordnung den Wert von Φ_5 (oder alternativ von Φ_4) verringert, wird sie beibehalten, sonst die Zuordnung zu einer anderen Teilmenge $\omega_{\lambda'}$ versucht. Diese Schritte werden wiederholt für alle möglichen Neuzuordnungen und alle Muster aus der Stichprobe. Sie werden iteriert, bis sich nichts mehr an der Zuordnung der Muster ändert. Das Prinzip zeigt Bild 4.8.3; dort wird die zu optimierende Funktion allgemein als Φ bezeichnet, kann also Φ_4 oder Φ_5 sein.

Die Effizienz wird dadurch verbessert, dass die Neuberechnung von Φ_5 nur die Veränderungen erfasst, also iterativ erfolgt. Sei bei der aktuellen Zuordnung von Mustern der Wert von Φ_5 durch (4.8.44) gegeben. Wird ein Muster ${}^{\varrho}\boldsymbol{c} \in \omega_\kappa$ aus ω_κ entfernt und nach ω_λ gebracht, so ergibt sich der veränderte Wert Φ'_5 aus dem alten Wert Φ_5 zu

$$\begin{aligned}
\Phi'_5 &= \frac{N_\kappa N_\lambda}{(N_\kappa - 1)(N_\lambda + 1)} \Phi_5 \\
&\quad + \alpha \left(\sum_j \chi({}^j\boldsymbol{c}, {}^{\varrho}\boldsymbol{c}_\lambda) g_0({}^j\boldsymbol{c} - {}^{\varrho}\boldsymbol{c}_\lambda) + \sum_k \chi({}^{\varrho}\boldsymbol{c}_\lambda, {}^k\boldsymbol{c}) g_0({}^{\varrho}\boldsymbol{c}_\lambda - {}^k\boldsymbol{c}) \right. \\
&\quad \left. - \sum_j \chi({}^j\boldsymbol{c}, {}^{\varrho}\boldsymbol{c}_\kappa) g_0({}^j\boldsymbol{c} - {}^{\varrho}\boldsymbol{c}_\kappa) - \sum_k \chi({}^{\varrho}\boldsymbol{c}_\kappa, {}^k\boldsymbol{c}) g_0({}^{\varrho}\boldsymbol{c}_\kappa - {}^k\boldsymbol{c}) \right) , \qquad (4.8.45) \\
\alpha &= \frac{1}{2N_1 N_2 \ldots N_{\kappa-1}(N_\kappa - 1)N_{\kappa+1} \ldots N_{\lambda-1}(N_\lambda + 1)N_{\lambda+1} \ldots N_k} .
\end{aligned}$$

Statt der Doppelsumme in (4.8.44) sind hier also nur einfache Summen auszuwerten. Natürlich ist eine analoge Vorgehensweise auch bei den Maßen in (4.8.26) möglich. Da χ und g_0 symmetrische Funktionen sind, lässt sich der Rechenaufwand in (4.8.45) weiter reduzieren, was in obiger Gleichung nicht explizit gemacht wurde, um den direkten Bezug zur Ausgangsgleichung

(4.8.44) zu behalten. Der verbleibende Parameter σ in (4.8.37) ist empirisch zu wählen. Er hängt von der mittleren Distanz der Merkmalsvektoren in der Stichprobe ab.

Hierarchische Zerlegungen

Um einen günstigen Kompromiss zwischen der Zahl der Klassen und der Homogenität der Muster in einer Klasse zu finden, eignen sich *hierarchische Zerlegungen*, die ähnlich wie in Bild 4.6.2b eine Folge zunehmend verfeinerter Zerlegungen liefern. Unter einer **Hierarchie** H von Zerlegungen wird eine Folge von $(m+1)$ Zerlegungen $A^0, A^1, \ldots, A^m$ der Stichprobe ω verstanden, wobei

$$\begin{aligned} A^0 &= \{\{{}^1\boldsymbol{c}\}, \{{}^2\boldsymbol{c}\}, \ldots, \{{}^N\boldsymbol{c}\}\} , \\ A^m &= \{\omega\} \end{aligned} \tag{4.8.46}$$

ist. Die Zerlegung A^0 enthält N Klassen mit je einem Muster, die Homogenität jeder Klasse ist maximal; dagegen enthält A^m nur eine Klasse mit N Mustern, die Homogenität dieser Klasse ist minimal. Weiterhin sei $A^{\nu-1}$ eine feinere Zerlegung als A^ν, $\nu = 1, \ldots, m$, d. h., dass die Klassen aus A^ν immer durch *Vereinigung* von zwei oder mehr Klassen aus $A^{\nu-1}$ entstehen. Die zu einer Zerlegung A^ν gehörigen Klassen seien disjunkt. Die Hierarchie H besteht also aus Teilmengen $\omega_1, \omega_2, \ldots, \omega_l$ von ω mit den oben angegebenen Eigenschaften. Ein Maß h zur Bewertung einer Hierarchie H ist eine für alle Teilmengen ω_λ, $\lambda = 1, \ldots, l$ definierte Funktion, die den Bedingungen

$$\begin{aligned} h(\omega_\lambda) &\geq 0 , \\ \omega_\kappa \subset \omega_\lambda &\Rightarrow h(\omega_\kappa) < h(\omega_\lambda) \end{aligned} \tag{4.8.47}$$

genügt. Beispiele für solche Funktionen sind

$$\begin{aligned} h(\omega_\lambda)_1 &= \max_{{}^j\boldsymbol{c}, {}^k\boldsymbol{c} \in \omega_\lambda} d_{jk} , \\ h(\omega_\lambda)_2 &= \sum_{j,k} d_{jk} , \\ h(\omega_\lambda) &= \sum_j ({}^j\boldsymbol{c} - \boldsymbol{\mu})^2 , \end{aligned} \tag{4.8.48}$$

wobei d_{jk} ein Abstandsmaß für zwei Muster ${}^j\boldsymbol{c}$, ${}^k\boldsymbol{c}$, z. B. gemäß (4.2.146), S. 355, ist. Damit lässt sich eine Hierarchie anschaulich als **Dendrogramm** wie in Bild 4.8.4 darstellen. In einer Stichprobe werden Muster in irgendeiner Reihenfolge angeordnet sein; das bedeutet, dass i. Allg. *nicht* in der Stichprobe benachbarte Muster zusammengefasst werden und sich eine kreuzungsfreie Darstellung erst nach einer Umordnung ergibt.

Hierarchien lassen sich im Wesentlichen auf zwei Arten konstruieren. Bei den agglomerativen oder "bottom–up" Verfahren beginnt man mit A^0, also der feinsten Zerlegung und fasst schrittweise Klassen zu übergeordneten Klassen zusammen, bis man bei A^m endet. Die divisiven oder "top–down" Verfahren beginnen mit A^m und zerlegen solange Klassen in homogenere Unterklassen, bis A^0 erreicht ist. Da die letzteren Verfahren mehr Rechenaufwand verursachen, wird hier nur das Prinzip der agglomerativen Konstruktion in Bild 4.8.5 gezeigt.

Damit die gefundene Hierarchie die Struktur der Stichprobe gut wiedergibt, sollte bei der Bildung neuer Klassen der Zuwachs an Inhomogenität möglichst klein sein. Durch unterschied-

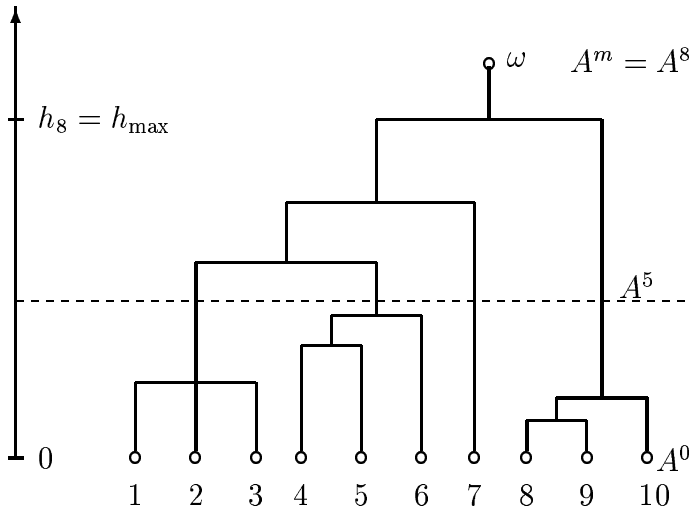


Bild 4.8.4: Die Veranschaulichung einer Hierarchie von Zerlegungen durch ein Dendrogramm

man wähle ein Maß $h(\omega_\kappa)$ und setze $A^0 = \{\omega_1, \omega_2, \ldots, \omega_N\}$ mit $\omega_j = \{{}^j\boldsymbol{c}\}$
im ν-ten Schritt bestimme man die zwei ähnlichsten Klassen $\omega_\kappa, \omega_\lambda \in A^{\nu-1}$, d.h. mit kleinstem Wert $h_\nu = h(\omega_{\kappa\lambda})$, und setze $\omega_{\kappa\lambda} = \{\omega_\kappa \cup \omega_\lambda\}$
die neue Zerlegung A^ν enthält alle Klassen von $A^{\nu-1}$, außer ω_κ und ω_λ, zuzüglich $\omega_{\kappa\lambda}$
UNTIL $A^m = \{\omega\}$

Bild 4.8.5: Prinzip der agglomerativen Bildung einer Hierarchie von Musterklassen

liche Wahl von h ergeben sich unterschiedliche Algorithmen. Übliche Maße $h(\omega_{\sigma\tau})$ zur Bewertung der Inhomogenität zweier Klassen $\omega_\sigma, \omega_\tau$ sind

$$\begin{aligned}
h(\omega_{\sigma\tau}) &= \min_{\{{}^j\boldsymbol{c}\in\omega_\sigma, {}^k\boldsymbol{c}\in\omega_\tau\}} d_{jk} && (\text{“single linkage”}) ,\\
h(\omega_{\sigma\tau}) &= \max_{\{{}^j\boldsymbol{c}\in\omega_\sigma, {}^k\boldsymbol{c}\in\omega_\tau\}} d_{jk} && (\text{“complete linkage”}) ,\\
h(\omega_{\sigma\tau}) &= \frac{1}{N_\kappa N_\lambda} \sum_{{}^j\boldsymbol{c}\in\omega_\sigma} \sum_{{}^k\boldsymbol{c}\in\omega_\tau} d_{jk} && (\text{“average linkage”}) ,\\
h(\omega_{\sigma\tau}) &= (\boldsymbol{\mu}_\sigma - \boldsymbol{\mu}_\tau)^2 .
\end{aligned} \tag{4.8.49}$$

Während die ersten drei Maße h (4.8.47) genügen, ist das beim letzten nicht der Fall. Im obigen Algorithmus setzt man

$$\boxed{h_\nu = h(\omega_{\kappa\lambda}) = \min_{\sigma,\tau} h(\omega_{\sigma\tau}) ,} \tag{4.8.50}$$

d. h. es werden die Klassen $\omega_\kappa, \omega_\lambda$ vereinigt, die den kleinsten Wert von h ergeben. Es kann sein, dass es mehrere Klassen $\omega_{\lambda i},\ i = 1, \ldots, k_\nu$ gibt, die von einer Klasse ω_κ den gleichen kleinsten Abstand haben. Um eine eindeutige Hierarchie zu erhalten, vereinigt man alle k_ν Klassen $\omega_{\lambda i}$ mit ω_κ. Die so gewonnene Hierarchie lässt sich, wie erwähnt, in einem Dendrogramm grafisch darstellen, wofür Bild 4.8.4 ein Beispiel ist. Ein möglicher Kompromiss zwischen Homogenität der Klassen und Zahl der Klassen ist durch die gestrichelte Linie angedeutet.

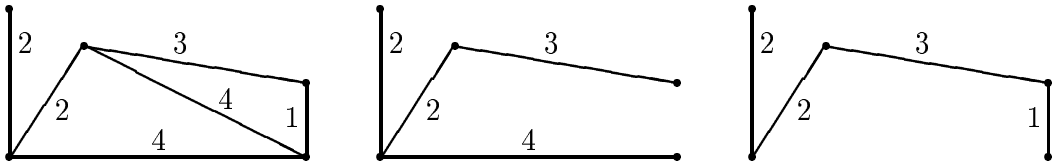


Bild 4.8.6: Das Bild zeigt links einen zusammenhängenden Graphen mit Schleifen, in der Mitte einen Verbindungsbaum mit dem Gewicht 11, rechts den Minimalbaum mit dem Gewicht 8

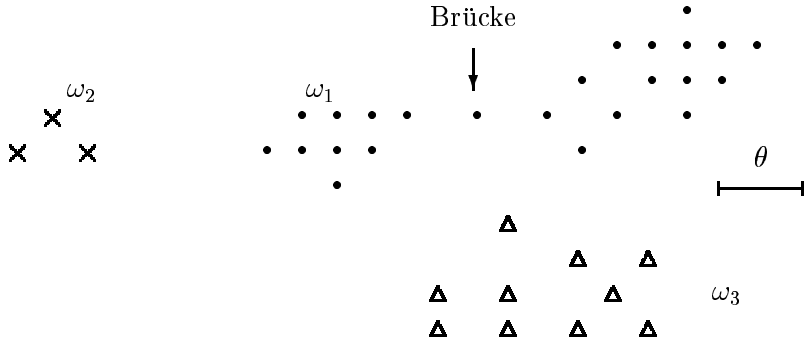


Bild 4.8.7: Zerlegung einer Stichprobe durch Eliminierung aller Kanten, deren Gewicht größer als θ ist, aus dem Graphen

4.8.5 Graphentheoretische Ansätze

Bei den graphentheoretischen Verfahren wird die Stichprobe ω einem gewichteten *Graphen* $\mathcal{G} = \{K, E, W\}$ zugeordnet, dessen Menge von Knoten $K = \omega$ ist, d. h. jedes Muster in der Stichprobe ist ein Knoten im Graphen, dessen Menge von Kanten $E = \omega \times \omega$ ist, d. h. jeder Knoten ist mit jedem anderen durch eine Kante verbunden, und dessen Menge von den Kanten zugeordneten Gewichten $W = \{d_{jk} = d({}^j\boldsymbol{c}, {}^k\boldsymbol{c}) \,|\, j, k = 1, \ldots, N\}$ ist, wobei $d({}^j\boldsymbol{c}, {}^k\boldsymbol{c})$ z. B. der EUKLID-Abstand der beiden Merkmalsvektoren ist. Zu diesem Graphen kann ein Verbindungsbaum konstruiert werden, der Pfade zwischen allen Knoten, aber keine Schleifen, besitzt. Das Gewicht so eines Baumes ist die Summe seiner Kantengewichte. Der Verbindungsbaum mit minimalem Gewicht ist der Minimalbaum oder der minimale Spannbaum ("minimal spanning tree") $\mathcal{B}_{\min}$. Bild 4.8.6 zeigt ein einfaches Beispiel. Graphentheoretische Verfahren zur Bestimmung von Häufungsgebieten entfernen nach heuristischen Kriterien Kanten aus dem Graphen $\mathcal{G}$ oder dem Minimalbaum $\mathcal{B}_{\min}$, sodass disjunkte Teilgraphen oder –bäume entstehen, die den Klassen entsprechen. Ein Vorteil des Minimalbaumes ist, dass er nur N Kanten enthält, der Graph dagegen $N(N-1)/2$; dafür erfordert die Berechnung des Minimalbaumes zusätzlichen Aufwand.

Ein spezielles Verfahren ist die sogenannte "single–linkage"–Methode, bei der alle Kanten mit einem Gewicht $d_{jk} > \theta$ aus $\mathcal{G}$ entfernt werden. Je nach Wahl von θ zerfällt dadurch der Graph in mehrere nicht zusammenhängende Teilgraphen, die als Klassen interpretiert werden. Dabei kann es zu der in Bild 4.8.7 gezeigten Brückenbildung kommen, und es können Muster *einer* Klasse relativ weit voneinander entfernt sein.

Die Ermittlung von Häufungsgebieten aus dem Minimalbaum beruht z. B. auf der Eliminierung sog. „inkonsistenter" Kanten, d. h. von Kanten, die ein signifikant höheres Gewicht haben als die in ihrer Umgebung. Ein Beispiel ist die Kante E_1 in Bild 4.8.8, deren Beseitigung die

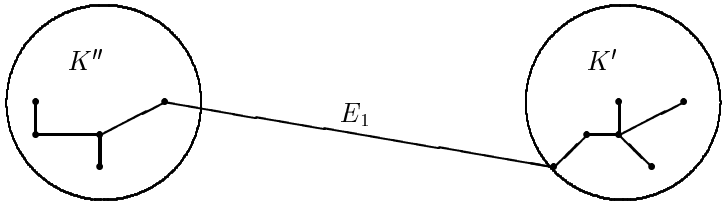


Bild 4.8.8: Ein Beispiel für eine „inkonsistente“ Kante E_1 in einem Minimalbaum

beiden disjunkten Teilmengen K' und K'' von Kanten ergibt. Im Folgenden werden zur Definition einer inkonsistenten Kante zwei Knoten $K_i, K_j \in K$ betrachtet; die zugehörige Kante E_{ij} hat das Gewicht d_{ij}. Man bestimmt die Menge der Kanten, die von K_i bzw. K_j in nicht mehr als δ Schritten erreichbar sind, *außer* der Kante E_{ij} von K_i nach K_j. Man bestimme die mittleren Gewichte $\mu_i(\delta), \mu_j(\delta)$ dieser Kanten sowie deren Streuungen $\sigma_i(\delta), \sigma_j(\delta)$. Die Kante E_{ij} wird als *inkonsistent* eliminiert, wenn einige oder alle der folgenden Bedingungen erfüllt sind

$$\begin{array}{lrcl} 1) & |d_{ij} - \mu_i(\delta)| & > & \gamma_1\, \sigma_i(\delta)\ , \\ 2) & |d_{ij} - \mu_j(\delta)| & > & \gamma_1\, \sigma_j(\delta)\ , \\ 3) & d_{ij} & > & \gamma_2\, \mu_i(\delta)\ , \\ 4) & d_{ij} & > & \gamma_2\, \mu_j(\delta)\ . \end{array} \tag{4.8.51}$$

Diese Prüfung auf Inkonsitenz wird für alle Kanten des Minimalbaumes durchgeführt. Das Ergebnis hängt ab von der Tiefe δ, von den Faktoren γ_1 und γ_2 sowie von der Kombination der vier Bedingungen, die zur Elimination führen. Diese Strategie eignet sich für deutlich getrennte Punktmengen (oder “Cluster”, Ballungsgebiete, Häufungsgebiete).

4.8.6 Bemerkungen

In diesem Kapitel wurden in den verschiedenen Abschnitten übergeordnete Ansätze zum Lernen bzw. zur automatischen Ermittlung von Klassenbereichen vorgestellt. Wegen der Fülle des veröffentlichten Materials konnten hier zu jedem übergeordneten Ansatz beispielhaft jeweils nur wenige spezielle Algorithmen und Gleichungen angegeben werden. Es stellt sich die Frage, welche Lernverfahren vorzuziehen sind. Eine gewisse Auswahl ergibt sich aus der Problemstellung, z. B. für überwachtes Lernen Verfahren von Abschnitt 4.2 – Abschnitt 4.5, für unüberwachtes Lernen Verfahren von Abschnitt 4.8.2 – Abschnitt 4.8.5. Trotzdem bleibt noch eine Vielzahl möglicher spezieller Lernalgorithmen. Auswahlkriterien können hier das Konvergenzverhalten und der Rechenaufwand sein. Leider ist das Konvergenzverhalten praktisch aller Verfahren bisher nur mangelhaft bekannt, und da das Konvergenzverhalten bei iterativen Algorithmen den Rechenaufwand bestimmt, ist auch dieser kaum vorherzusagen. Im Unterschied zu einem Konvergenzbeweis, der für $N \to \infty$ geführt wird, ist es für die praktische Anwendung eines Algorithmus entscheidend, ob man bei einem realistischen Problem eine akzeptable Klassifikatorleistung mit vertretbarem Rechenaufwand und einer begrenzten Stichprobe erzielen kann; die konkreten Anforderungen werden von Fall zu Fall unterschiedlich liegen.

Konvergenzbeweise sind für einige Algorithmen möglich und wurden jeweils erwähnt. Etwas übertrieben, aber durch die Übertreibung verdeutlichend, kann man sagen, dass ein Konvergenzbeweis für die praktische Anwendung eines Lernalgorithmus nicht unbedingt notwendig oder hinreichend ist. Er braucht nicht notwendig zu sein, weil es z. B. in kommerziellen Geräten

immer wieder rein heuristische Trainingsalgorithmen gibt, die offenbar erfahrungsgemäß die gewünschte Leistung erbringen. Er braucht nicht hinreichend zu sein, weil die zu einem Konvergenzbeweis erforderlichen Voraussetzungen – z. B. $N \to \infty$, statistisch unabhängige Stichprobenelemente, bekannte parametrische Familie von Verteilungsdichten, bekannte Familie von Trennfunktionen, usw. – in einem praktischen Problem stets nur approximierbar sind. Wie sich die näherungsweise Einhaltung von Voraussetzungen auswirkt, ist theoretisch nicht überschaubar. Es gibt zu den meisten Lernalgorithmen veröffentlichte experimentelle Ergebnisse; danach konvergiert jeder der angegebenen Algorithmen, und insbesondere der EM–Algorithmus hat sich als überaus leistungsfähig erwiesen.

4.9 Objektklassifikation und -lokalisation (VA.1.1.2, 14.05.2004)

4.9.1 Übersicht

Das Problem der Erkennung dreidimensionaler Objekte in zweidimensionalen Ansichten wurde in Abschnitt 1.5 vorgestellt. In Abschnitt 4.2.1 wurde ein kurzer Überblick über einige Möglichkeiten der statistischen Modellierung von Beobachtungen gegeben, wobei in (4.2.30) – (4.2.34), S. 332, einige spezielle Ansätze bereits angedeutet wurden. Diese greifen wir im folgenden wieder auf, um zu genaueren Aussagen zu kommen. Ein sehr allgemeiner Ansatz ist die Angabe der – durch klassenspezifische Parameter $\boldsymbol{a}_\kappa$ und Lageparameter $\boldsymbol{\theta}$ bedingten – Verbundverteilung aller Tupel von unabhängigen und abhängigen Variablen eines Musters, also

$$
\begin{aligned}
p(\boldsymbol{f}\,|\,\boldsymbol{a}_\kappa,\,\boldsymbol{\theta}) &= p(\boldsymbol{f}\,|\,\Omega_\kappa, \boldsymbol{t}, \boldsymbol{R}) && (4.9.1)\\
&= p\left(\left\{(j,k,\tau,f_{jk\tau})^{\mathsf{T}}\right\}\,|\,\boldsymbol{a}_\kappa,\,\boldsymbol{t},\boldsymbol{R}\right)\ , && (4.9.2)\\
&\quad j=0,1,\ldots,M_x-1,\ k=0,1,\ldots,M_y-1,\\
&\quad \tau=0,1,\ldots,T-1,\ f_{jk\tau}=b_l\in\{1,\ldots,L\}\ .
\end{aligned}
$$

Dabei wurden die Parameter $\boldsymbol{\theta}$ auf Translations– und Rotationsparameter spezialisiert. Diese Gleichung entspricht (4.2.30), S. 332, wenn man die (kontinuierlichen oder diskreten) Positionen $\boldsymbol{x}$ durch die Indizes (j,k,τ) angibt und statt des Merkmalsvektors $\boldsymbol{c}_{\boldsymbol{x}}$ den Grauwert $f_{jk\tau}$ verwendet.

Wegen ihrer Allgemeinheit ist diese Gleichung, zumindest beim gegenwärtigen Stand der Technik, nicht brauchbar. Vereinfachungen ergeben sich durch eine oder mehrere der folgenden Spezialisierungen:

- Annahme statistischer Unabhängigkeit der Tupel, bzw. von Abhängigkeiten begrenzter Ordnung.
- Verwendung kontinuierlicher statt diskreter Zufallsvariablen.
- Verwendung marginaler Verteilungsdichten.
- Statt der Grauwerte f eines Musters werden daraus extrahierte Merkmale verwendet, die sowohl numerischer als auch symbolischer Art (vgl. Abschnitt 3.1) sein können.
- Die Verbundverteilung (4.9.1) kann auf unterschiedliche Arten faktorisiert werden, wie es in (4.2.33) und (4.2.34), S. 332, bereits angedeutet wurde.
- Die Abhängigkeit vom Zeitindex τ wird fallengelassen.
- Die Parameter Translation $\boldsymbol{t}$ und Rotation $\boldsymbol{R}$ werden eingeschränkt, z. B. auf Bewegungen in einer Ebene.
- Statt allgemeiner Objekte im $\mathbb{R}^3$ werden nur zweidimensionale Ansichten von Objekten zugelassen.

Daraus resultieren zahlreiche Varianten für statistische Objektmodelle, von denen einige betrachtet werden.

In diesem Abschnitt werden nur stochastische Verfahren zur Erkennung und bei Bedarf Lokalisation von Objekten betrachtet. Für andere wird auf die Literatur verwiesen. Speziell wird auf histogrammbasierte Ansätze eingegangen, die dann vorteilhaft sind, wenn eine Lokalisation nicht erforderlich ist; weiter werden Verfahren vorgestellt, die *lokale Merkmale* an bestimmten Bildpositionen berechnen und auch eine Lokalisation erlauben; schließlich wird gezeigt, dass auch unter Verwendung der Ergebnisse einer Segmentation klassifiziert und lokalisiert werden kann.

Die prinzipielle Vorgehensweise besteht darin, zunächst eine geeignete Form für die klassenbedingte Verteilungsdichte $p(\boldsymbol{f}\,|\,\Omega_\kappa, \boldsymbol{t}, \boldsymbol{R})$ in (4.9.1) zu bestimmen. Statt der Abtastwerte $\boldsymbol{f}$ werden in der Regel lokale oder globale Merkmale verwendet, was hier aber noch keine Rolle spielt. Die klassenspezifischen Parameter $\boldsymbol{a}_\kappa$ werden mit einer Stichprobe von Bildern, die Objekte aus der Klasse Ω_κ enthalten und bei denen – falls eine Lokalisation erforderlich ist – auch die Lageparameter der Objekte bekannt sind, bestimmt. Ein Maximum-likelihood-Schätzwert ist

$$\boxed{\widehat{\boldsymbol{a}}_\kappa = \underset{\boldsymbol{a}_\kappa}{\operatorname{argmax}}\, p\left({}^1\boldsymbol{f}_\kappa, \ldots, {}^N\boldsymbol{f}_\kappa \,|\, \boldsymbol{a}_\kappa, {}^1\boldsymbol{\theta}_\kappa, \ldots, {}^N\boldsymbol{\theta}_\kappa\right) \;, \quad \kappa = 1, \ldots, k \;.} \tag{4.9.3}$$

Wenn ein Bild $\boldsymbol{f}$ mit einem Objekt unbekannter Klasse und Lage beobachtet wird, wird für jede Klasse die am besten passende Lage des Objektes berechnet aus

$$\boxed{\widehat{\boldsymbol{\theta}}_\kappa = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{f}\,|\,\widehat{\boldsymbol{a}}_\kappa, \boldsymbol{\theta}) \;, \quad \kappa = 1, \ldots, k \;.} \tag{4.9.4}$$

Mit den je Klasse berechneten Lageparametern wird dann mit (4.1.35), S. 315, bzw. Satz 4.3, S. 315, das Objekt klassifiziert gemäß

$$\boxed{\kappa = \underset{\lambda}{\operatorname{argmax}}\, p(\boldsymbol{f}\,|\,\widehat{\boldsymbol{a}}_\lambda, \widehat{\boldsymbol{\theta}}_\lambda) \;.} \tag{4.9.5}$$

In diesem Zusammenhang betrachten wir auch das **Wiederfinden von Bildern** oder Objekten in großen Datenbeständen (“image retrieval”).

4.9.2 Lageunabhängige Erkennung mit Histogrammen

Das (normierte) Grauwerthistogramm $\mathcal{H}(l\,|\,f)$ wurde bereits in (2.2.6), S. 79, eingeführt. Im Prinzip können statt des Grauwertes f auch die Farbwerte $f_\mathrm{r}, f_\mathrm{g}, f_\mathrm{b}$ oder der Wert $c_\nu = T_\nu\{f\}$ irgendeiner lokalen Operation, speziell z. B. einer linearen Filterung genommen werden. Die lokale Operation liefert, angewendet an verschiedenen Positionen im Bild, für die ν-te Operation eine Menge $\widetilde{C} = \{\boldsymbol{c}_{m,\nu}\}$, $m = 1, \ldots, N_c$ von Merkmalen, wie in Abschnitt 3.7.2 bereits beschrieben.

Wenn man den Wertebereich $c_{\nu,\min} \leq c_\nu \leq c_{\nu,\max}$ des Merkmals in L_ν Stufen $b_{l_\nu}, l_\nu = 1, \ldots, L_\nu$ quantisiert, lässt sich das (normierte) *Histogramm* der Werte von c_ν bei Beobachtung eines Objektes aus der Klasse Ω_κ berechnen zu

$$\mathcal{H}(l_\nu\,|\,c_\nu, \Omega_\kappa) = \frac{\sum\limits_{c_{m,\nu}=b_{l_\nu}\,|\,\boldsymbol{f}\in\Omega_\kappa} 1}{M_x M_y}, \quad l_\nu = 1, \ldots, L_\nu \;. \tag{4.9.6}$$

Statt $\mathcal{H}(l_\nu\,|\,c_\nu, \Omega_\kappa)$ wird auch die Bezeichnung $\mathcal{H}(l_\nu\,|\,c_\nu, \kappa)$ verwendet. Es werden $\nu = 1, \ldots, n$ Merkmale berechnet, für die – insbesondere bei großer Zahl von Merkmalen – die n eindimensionalen Histogramme mit (4.9.6) berechnet werden. Bei statistischer Unabhängigkeit ergibt sich mit (4.2.13), S. 326, das n-dimensionale Histogramm zu

$$\begin{aligned} \mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa) &= \mathcal{H}(l_1, \ldots, l_n\,|\,c_1, \ldots, c_n, \kappa) = \prod_{\nu=1}^{n} \mathcal{H}(l_\nu\,|\,c_\nu, \kappa) \\ &= \left[\mathcal{H}(l_1\,|\,c_1, \kappa) \cdot \ldots \cdot \mathcal{H}(l_\nu\,|\,c_\nu, \kappa) \cdot \ldots \cdot \mathcal{H}(l_n\,|\,c_n, \kappa)\right] \;. \end{aligned} \tag{4.9.7}$$

Für n Merkmale, die jeweils mit L_ν Werten quantisiert sind, erhält man also $L_1 \cdot L_2 \cdot \ldots \cdot L_n$ Tupel der angegebenen Form. Wenn die Zahl der Merkmale klein ist, z. B. $n = 2, 3$ oder 4, so kann statt der n eindimensionalen Histogramme auch ein n-dimensionales Histogramm $\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa)$ analog zu (4.9.6) berechnet werden. Bei einer großen Zahl von Merkmalen scheitert die Berechnung der mehrdimensionalen Histogramme am Aufwand („der Fluch der hohen Dimension“), wie schon in Abschnitt 4.2.1 ausgeführt wurde. Wenn man k Objekte klassifizieren will, werden als Referenzen deren n-dimensionale Histogramme aus einer Stichprobe von Bildern geschätzt. Damit liegt ein statistisches Modell der Objekte in Form einer nichtparametrischen Schätzung der bedingten Verteilungsdichten der Merkmale vor (s. auch Abschnitt 4.2.1 und Abschnitt 4.2.6). Zur Wahl der Zahl der diskreten Werte L_ν des Histogramms besteht ein Vorschlag darin, unabhängig von der Art der Operation $L_\nu = L = 32$ zu setzen. Zur Art und Zahl der konkret verwendeten lokalen Operationen wird auf Abschnitt 3.7.2 verwiesen.

Wenn ein neu beobachtetes Objekt zu klassifizieren ist, kann dieses zum einen durch einen Vergleich der Referenzhistogramme der Objkete mit dem Histogramm der neuen Beobachtung erfolgen, zum anderen durch direkten Rückgriff auf Satz 4.3, S. 315.

Zum Vergleich des Histogramms $\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa)$ eines Objektes aus Ω_κ mit dem Histogramm $\mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})$ der ϱ-ten neuen Beobachtung wurden unterschiedliche Maße für den **Abstand zweier Histogramme** vorgeschlagen. Der Abstand zwischen zwei n-dimensionalen Histogrammen erfordert stets eine Operation über alle $L_1 \cdot \ldots \cdot L_n$ Einträge oder Tupel im Histogramm, für die wir am Beispiel des *quadratischen Abstands* eine vereinfachte Notation einführen gemäß

$$\begin{aligned} d_{\mathcal{H},q}(\kappa, \varrho) &= d\left(\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa),\, \mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})\right) \\ &= \sum_{l_1=1}^{L_1} \ldots \sum_{l_\nu=1}^{L_\nu} \ldots \sum_{l_n=1}^{L_n} \left(\prod_{\nu=1}^{n} \mathcal{H}(l_\nu\,|\,c_\nu, \kappa) - \prod_{\nu=1}^{n} \mathcal{H}(l_\nu\,|\,{}^{\varrho}c_\nu) \right)^2 \\ &= \sum_{l_1,\ldots,l_n} \left(\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa) - \mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})\right)^2 . \end{aligned} \tag{4.9.8}$$

Weitere Abstandsmaße sind der χ^2-Abstand $d_{\mathcal{H},\chi}$, der KULLBACK-LEIBLER-Abstand $d_{\mathcal{H},\mathrm{KL}}$ und der BHATTACHARYYA-Abstand $d_{\mathcal{H},\mathrm{B}}$ zweier Histogramme

$$d_{\mathcal{H},\chi}(\kappa, \varrho) = \sum_{l_1,\ldots,l_n} \frac{\left(\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa) - \mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})\right)^2}{\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa) + \mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})} , \tag{4.9.9}$$

$$d_{\mathcal{H},\mathrm{KL}}(\kappa, \varrho) = \sum_{l_1,\ldots,l_n} \left(\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa) - \mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})\right) \log \left[\frac{\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa)}{\mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})} \right] , \tag{4.9.10}$$

$$d_{\mathcal{H},\mathrm{B}}(\kappa, \varrho) = 1 - \sum_{l_1,\ldots,l_n} \sqrt{\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa) \cdot \mathcal{H}(\boldsymbol{l}\,|\,{}^{\varrho}\boldsymbol{c})} . \tag{4.9.11}$$

Beim Rückgriff auf Satz 4.3 wird, wie schon generell in Kapitel 4.2, für einen im Muster ${}^{\varrho}\boldsymbol{f}$ an der Position m beobachten Merkmalsvektor $\boldsymbol{c}_m$ die bedingte Wahrscheinlichkeit $p(\boldsymbol{c}_m\,|\,\Omega_\kappa)$ berechnet. Der Wert dieser Wahrscheinlichkeit kann aus dem n-dimensionalen Histogramm $\mathcal{H}(\boldsymbol{l}\,|\,\boldsymbol{c}, \kappa)$ in (4.9.7) entnommen werden. Nun wird *ein* lokales Merkmal an irgendeiner Position m i. Allg. keine zurverlässige Klassifikation erlauben. Wie schon in Abschnitt 3.7.2 beschrieben, werden daher N_c lokale Merkmale an unterschiedlichen Positionen im Bild berechnet und ihre Wahrscheinlichkeiten dem Histogramm entnommen. Unter der Annahme statistischer Unabhängigkeit der lokalen Merkmale ist dann die bedingte Wahrscheinlichkeit der Beobachtung

einer Menge $\widetilde{C} = \{c_m\}$ von N_c Merkmalen

$$p\left(\{c_m\}|\Omega_\kappa\right) = \prod_{m=1}^{N_c} p(\boldsymbol{c}_m|\Omega_\kappa) \ . \tag{4.9.12}$$

Die Klasse ergibt sich bei Anwendung des BAYES-Klassifikators dann direkt aus (4.1.35), S. 315. Die Frage, an wievielen Positionen man Merkmale berchnen muss, ist experimentell zu entscheiden; je nach Problem wird eine Zahl N_c angegeben, die bei $1\% - 20\%$ der Bildpunkte liegt. Die oben geforderte statistische Unabhängigkeit der lokalen Merkmalsvektoren wird umso besser erfüllt sein, je weiter die Merkmalspositionen auseinander liegen, also je größer die Abstände Δx_c, Δx_c der Merkmalspositionen in (3.7.1), S. 218, sind. Wenn allerdings diese Abstände zu groß werden, erhält man nur wenige Merkmale c_m und damit eine hohe Fehlerrate, sodass ein Kompromiss experimentell zu bestimmen ist.

Histogramme von Merkmalen eines Bildes ergeben oft bereits sehr leistungsfähige Verfahren zur Objekterkennung. Mit diesem Vorgehen wird offensichtlich (weitgehende) Invarianz der bedingten Verteilungsdichte gegenüber der Translation und Rotation von Objekten in der Bildebene erreicht; aber auch Invarianz gegenüber jeder Permutation der Bildpunkte. Letztere Eigenschaft kann durch die Verwendung lokaler, d. h. in Datenfenstern geeigneter Größe berechneten, Histogrammen gemildert werden. Für die Lokalisation von Objekten ist dieser Ansatz also nicht geeignet, die Lageparameter wurden daher fortgelassen.

4.9.3 Klassifikation und Lokalisation mit lokalen Merkmalen

Im Folgenden wird ein Ansatz vorgestellt, der von ortsabhängigen lokalen Merkmalen wie in Abschnitt 3.7.2 und (3.7.7), S. 219, ausgeht, Translationen und Rotationen des Objektes sowohl in der Bildebene als auch aus der Bildebene heraus in die stochastische Modellierung einbezieht, Objekt und Hintergrund stochastisch modelliert und eine Klassifikation und Lokalisation des Objektes ermöglicht.

Lokalisationsparameter und Objektfenster

Für die Objektlokalisation, d. h. die Bestimmung der drei Translationsparameter $(t_x,\, t_y,\, t_z)$ und der drei Rotationsparameter $(\theta_x,\, \theta_y,\, \theta_z)$ werden die *internen* Parameter $\boldsymbol{\theta}_{\text{int}} = (t_x,\, t_y,\, \theta_z)$ für Bewegungen *innerhalb* der Bildebene und die *externen* Parameter $\boldsymbol{\theta}_{\text{ext}} = (\theta_x,\, \theta_y,\, t_z)$ für Bewegungen *aus* der Bildebene heraus unterschieden; die Parameter zeigt Bild 4.9.1 (links). Bewegungen in der Bildebene lassen Objektform und -größe unverändert, Bewegungen aus der Bildebene heraus verändern sie, wie auch Bild 4.9.1 (rechts) zeigt.

Ein Objekt nimmt in einem aufgenommenen Bild i. Allg. nur einen Teil des Bildes ein, der Rest ist *Hintergrund*, der homogen sein kann wie in Bild 4.9.1 (rechts) oder inhomogen. In jedem Fall ist es für die Klassifikation und Lokalisation störend, wenn lokale Merkmale über das ganze Bild berechnet werden und nicht nur in dem Teil, der von dem Objekt eingenommen wird. Daher wird ein *Objektfenster* A eingeführt, das nicht notwendig rechteckig sein muss und den vom Objekt eingenommenen Bereich im Bild kennzeichnet. Das Objektfenster kann konstante Form haben, wie es insbesondere bei Bewegungen innerhalb der Bildebene hinreichend und in Bild 4.9.2 gezeigt ist. Bei Bewegungen aus der Bildebene heraus ist ein variables Objektfenster zweckmäßig; darauf wird unten mit Bild 4.9.3 eingegangen.

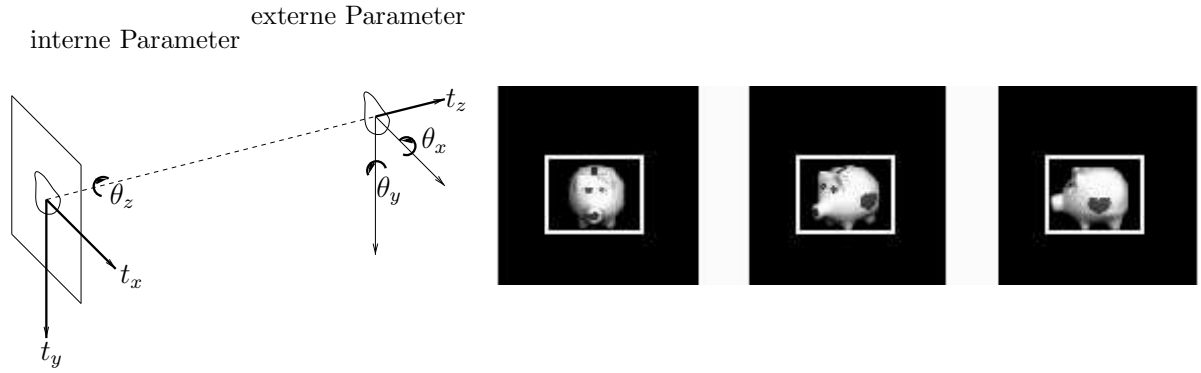


Bild 4.9.1: Interne und externe Parameter für die Objektlokalisation (links); Wirkung externer Rotation auf Objektform und -größe (rechts)

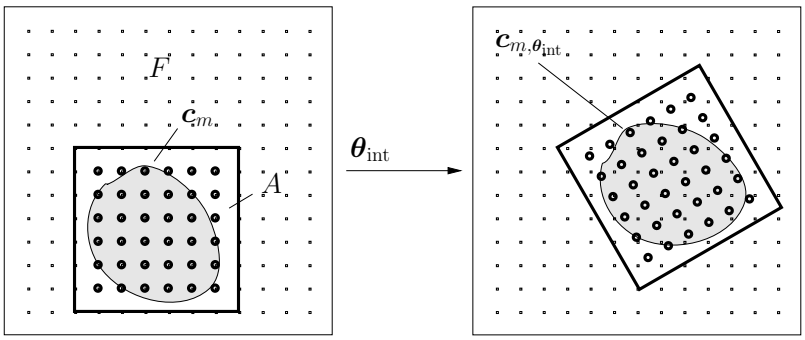


Bild 4.9.2: Rechteckiges Objektfenster mit konstanter Form und Größe; es kann in der Bildebene verschoben und gedreht werden mit den internen Parametern $\boldsymbol{\theta}_{\text{int}}$

Statistisches Modell mit internen Parametern

Es werden lokale Merkmale $\boldsymbol{c}(\boldsymbol{x}_m) = \boldsymbol{c}_m$, wie in Abschnitt 3.7.2 beschrieben, an diskreten Positionen $\boldsymbol{x}_m$ berechnet, und zwar zunächst im gesamten Bild. Sie werden in einer Menge $\widetilde{C}$ zusammengefasst. Wenn man annimmt, dass die Merkmalsvektoren $\boldsymbol{c}_m$, $m = 1, \ldots, N_c$ statistisch unabhängig sind und dass die Merkmale außerhalb des Objektfensters A annähernd gleichverteilt sind, erhält man für ihre Verteilungsdichte je Klasse

$$p(\widetilde{C}|\Omega_\kappa) = \prod_{\boldsymbol{c}_m \in A} p(\boldsymbol{c}_m|\Omega_\kappa) \prod_{\boldsymbol{c}_m \notin A} p(\boldsymbol{c}_m) \approx \prod_{\boldsymbol{c}_m \in A} p(\boldsymbol{c}_m|\Omega_\kappa) \; . \tag{4.9.13}$$

Wenn nichts über die Verteilung der Merkmale außerhalb des Objektfensters A, also im Hintergrund, bekannt ist, ist die Gleichverteilung eine plausible Annahme.

Eine interne Rotation um den Winkel θ_z, erfasst in der Rotationsmatrix $\boldsymbol{R}(\theta_z) = \boldsymbol{R}_{\text{int}}$ und eine interne Translation mit dem Vektor $\boldsymbol{t}_{\text{int}} = (t_x, t_y)$ bewirkt eine Transformation der Position $\boldsymbol{x}_m$ der Merkmale zu $\boldsymbol{x} = \boldsymbol{R}_{\text{int}}\boldsymbol{x}_m + \boldsymbol{t}_{\text{int}}$. Damit ergibt sich die Verteilungsdichte der Merkmalsvektoren nach interner Transformation zu

$$\begin{aligned} p(\widetilde{C}|\Omega_\kappa) &= \prod_{\boldsymbol{x}_m \in A} p\left(\boldsymbol{c}(\boldsymbol{x})|\Omega_\kappa, \boldsymbol{x} = \boldsymbol{R}_{\text{int}}\boldsymbol{x}_m + \boldsymbol{t}_{\text{int}}\right) \\ &= \prod_{\boldsymbol{x}_m \in A} p\left(\boldsymbol{c}|\Omega_\kappa, \boldsymbol{x}_m, \boldsymbol{R}_{\text{int}}, \boldsymbol{t}_{\text{int}}\right) = \prod_{\boldsymbol{x}_m \in A} p\left(\boldsymbol{c}|\Omega_\kappa, \boldsymbol{x}_m, \theta_z, (t_x, t_y)\right) \; . \end{aligned} \tag{4.9.14}$$

Wenn man annimmt, dass jeder Merkmalsvektor $\boldsymbol{c}(\boldsymbol{x})$ normalverteilt ist mit Mittelwert $\boldsymbol{\mu}_{\kappa,m}$ und Kovarianzmatrix $\boldsymbol{\Sigma}_{\kappa,m}$, ergibt sich für die Verteilungsdichte der lokalen Merkmalsvektoren

$$\boxed{\begin{aligned} p(\widetilde{C}|\Omega_\kappa) &= \prod_{\boldsymbol{x}_m \in A} \mathcal{N}\left(\boldsymbol{c}|\boldsymbol{x}_m, \boldsymbol{\mu}_{\kappa,m}, \boldsymbol{\Sigma}_{\kappa,m}, \theta_z, (t_x, t_y)\right) \\ &= \prod_{\boldsymbol{x}_m \in A} \mathcal{N}\left(\boldsymbol{c}(\boldsymbol{R}_{\text{int}}\boldsymbol{x}_m + \boldsymbol{t}_{\text{int}})\,|\,\boldsymbol{\mu}_{\kappa,m}, \boldsymbol{\Sigma}_{\kappa,m}\right) \ . \end{aligned}} \tag{4.9.15}$$

In diesem statistischen Modell sind die klassenspezifischen Eigenschaften eines Objektes in den Parametern $\boldsymbol{\mu}_{\kappa,m}, \boldsymbol{\Sigma}_{\kappa,m}$ enthalten, die Objektlage (in der Bildebene) in den Parametern $\theta_z, (t_x, t_y)$ bzw. $\boldsymbol{R}_{\text{int}}, \boldsymbol{t}_{\text{int}}$.

Statistisches Modell mit externen Parametern

Der Ansatz zur Berücksichtigung externer Parameter besteht darin, das obige Modell (4.9.15) als Basis zu verwenden, aber die klassenspezifischen Parameter nun als *Funktionen* der externen Parameter darzustellen. Damit ergibt sich

$$(\boldsymbol{\mu}_{\kappa,m}, \boldsymbol{\Sigma}_{\kappa,m}) = \left(\boldsymbol{\mu}_{\kappa,m}(\theta_x, \theta_y, t_z),\ \boldsymbol{\Sigma}_{\kappa,m}(\theta_x, \theta_y, t_z)\right) \ . \tag{4.9.16}$$

Die (unbekannten) Funktionen werden durch eine Entwicklung nach Basisfunktionen $\varphi_r(\theta_x, \theta_y, t_z),\ r = 0, 1, \ldots, L_r$ approximiert. Wenn man die einschränkende Annahme macht, dass nur eine externe Rotation θ_y möglich ist, die Kovarianzmatrizen Diagonalmatrizen mit konstanten Koeffizienten sind und daher nur jede Komponente des Mittelwertsvektors durch eine Reihenentwicklung nach Basisfunktionen $\varphi_r(\theta_y)$ approximiert wird, sind geschlossene Lösungen für die Maximum-likelihood-Schätzwerte der unbekannten Parameter möglich. In diesem Fall gilt also

$$\begin{aligned} \theta_x &= \text{const}\ , \quad t_z = \text{const}\ , \\ \boldsymbol{\Sigma}_{\kappa,m} &= \text{diag}(\sigma_{\kappa,m,1}, \ldots, \sigma_{\kappa,m,n})\ , \quad \sigma_{\kappa,m,\nu} = \text{const}\ , \\ \mu_{\kappa,m,\nu}(\theta_y) &= \sum_{r=0}^{L_\mu} a_{\kappa,m,\nu,r}\varphi_r(\theta_y)\ . \end{aligned} \tag{4.9.17}$$

Wählt man als Basisfunktionen Potenzen von θ_y, so erhält man speziell

$$\mu_{\kappa,m,\nu}(\theta_y) = \sum_{r=0}^{L_\mu} a_{\kappa,m,\nu,r}\theta_y^r\ . \tag{4.9.18}$$

Im Prinzip können andere Basisfunktionen verwendet werden, z. B. trigonometrische Funktionen, worauf hier jedoch nicht weiter eingegangen wird..

Mit diesen Annahmen geht (4.9.15) über in

$$\boxed{\begin{aligned} p(\widetilde{C}|\Omega_\kappa) &= \prod_{\boldsymbol{x}_m \in A} \prod_{\nu=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2_{\kappa,m,\nu}}} \exp\left[-\frac{b}{2\sigma^2_{\kappa,m,\nu}}\right]\ , \\ b &= \left(c_\nu(\boldsymbol{R}_{\text{int}}\boldsymbol{x}_m + \boldsymbol{t}_{\text{int}}) - \sum_{r=0}^{L_\mu} a_{\kappa,m,\nu,r}\theta_y^r\right)^2\ . \end{aligned}} \tag{4.9.19}$$

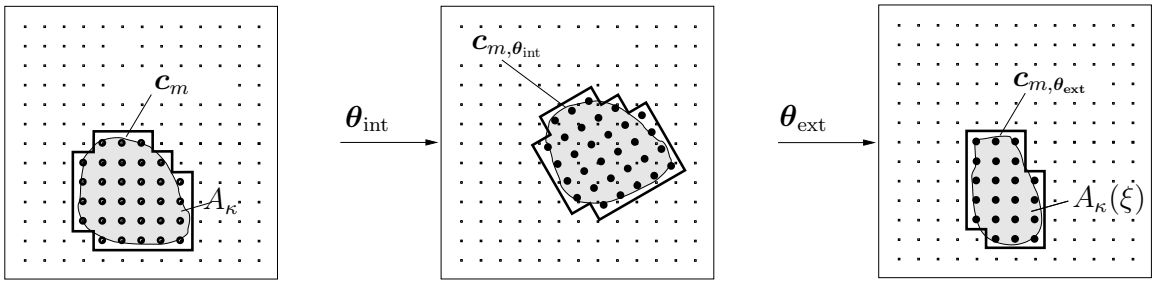


Bild 4.9.3: An das Objekt angepasstes Objektfenster mit konstanter Form und Größe für Bewegungen in der Bildebene und *variabler* Form und Größe für Bewegungen aus der Bildebene heraus

Zur Abkürzung wird für diesen Ausdruck auch geschrieben

$$\begin{aligned} p(\widetilde{C}|\Omega_\kappa) &= p\left(\widetilde{C}_A|\boldsymbol{a}_\kappa, \boldsymbol{R}, \boldsymbol{t}\right) \\ &= \prod_{\boldsymbol{x}_m \in A} \mathcal{N}\left(\boldsymbol{c}|\boldsymbol{x}_m, \{a_{\kappa,m,\nu,r}, \sigma_{\kappa,m,\nu}\}, \theta_{y,\text{ext}}, (\theta_z, t_x, t_y)_{\text{int}}\right) \ . \end{aligned} \tag{4.9.20}$$

Mit der Bezeichnung $\widetilde{C}_A$ wird zum Ausdruck gebracht, dass lokale Merkmale nur im Objektfenster A berechnet werden; die klassenspezifischen Parameter sind in $\boldsymbol{a}_\kappa = \{a_{\kappa,m,\nu,r}, \sigma_{\kappa,m,\nu}\}$ zusammengefasst; die Lageparameter sind in Rotations- und Translationsparameter $\boldsymbol{R}$ und $\boldsymbol{t}$ bzw. in externe und interne Parameter $\theta_{y,\text{ext}}$ und $(\theta_z, t_x, t_y)_{\text{int}}$ getrennt. Zur Schätzung der Parameter wird auf die in Abschnitt 4.11 zitierte Literatur verwiesen.

In (4.9.20) wird das Produkt über die N_A Normalverteilungen der Merkmalsvektoren im Objektfenster A gebildet. Wie Bild 4.9.3 zeigt, ist die Zahl N_A i. Allg. *nicht* konstant, sondern von den externen Parametern und damit von der Klasse abhängig; statt N_A ist also eine klassenspezifische Zahl N_{A_κ} zu verwenden. Daher ist eine Normierung der Wahrscheinlichkeiten zur Berücksichtigung der nicht konstanten Zahl von Merkmalsvektoren zweckmäßig. In experimentellen Untersuchungen hat sich das geometrische Mittel als geeignet erwiesen, sodass statt (4.9.20) der Ausdruck

$$p_{\text{gm}}(\widetilde{C}|\Omega_\kappa) = \sqrt[N_{A_\kappa}]{p\left(\widetilde{C}_{A_\kappa}|\boldsymbol{a}_\kappa, \boldsymbol{R}, \boldsymbol{t}\right)} \tag{4.9.21}$$

zur Klassifikation verwendet wird.

Ermittlung des Objektfensters

Insbesondere zur Erfassung externer Transformationen ist ein Objektfenster konstanter Größe weniger geeignet, da sich die Objektgröße bei Bewegungen aus der Bildebene heraus stark verändern kann und damit viele Merkmale berechnet würden, die nicht zum Objekt gehören. Das reduziert die Erkennungsrate. Daher wird ein Objektfenster eingeführt, das an die Objektgröße angepasst wird, wie es Bild 4.9.3 zeigt.

Bei internen Transformationen bleibt die Größe des Objektfensters konstant bei externen wird sie durch eine Funktion $\xi_m(\boldsymbol{\theta}_{\text{ext}})$ angepasst, indem jeder Merkmalsvektor $\boldsymbol{c}_m$ entweder dem Objekt oder dem Hintergrund zugewiesen wird. Mit Basisfunktionen $\varphi_\xi(\boldsymbol{\theta}_{\text{ext}})$ wird die

Bild 4.9.4: Beispiele für die Objekterkennung und -lokalisation; oben: nur interne Transformationen (zweidimensionaler Fall); unten: interne und externe Transformationen (dreidimensionaler Fall)

(unbekannte) Funktion ξ_m approximiert durch

$$\xi_m(\boldsymbol{\theta}_{\text{ext}}) = \sum_{r=0}^{N_\xi - 1} a_{\xi,m,r} \varphi_\xi(\boldsymbol{\theta}_{\text{ext}}) \; . \tag{4.9.22}$$

Die unbekannten Koeffizienten $a_{\xi,m,r}$ werden bestimmt, indem man die Funktion

$$\widetilde{\xi}_m(\boldsymbol{\theta}_{\text{ext}}) = \begin{cases} 0 & : \quad \boldsymbol{c}_m(\boldsymbol{\theta}_{\text{ext}}) < \theta_\xi \\ 1 & : \quad \boldsymbol{c}_m(\boldsymbol{\theta}_{\text{ext}}) \geq \theta_\xi \end{cases} \tag{4.9.23}$$

mit Hilfe von Trainingsbildern durch ξ approximiert. Der Schwellwert θ_ξ wird manuell bestimmt. Zur Schätzung der klassenspezifischen Parameter sowie zur Objekterkennung werden nur Merkmale aus dem Objektfenster A_κ verwendet, d. h. Merkmale $\boldsymbol{c}_m$, für die $\widetilde{\xi}_m(\boldsymbol{\theta}_{\text{ext}}) = 1$ gilt. Das Objektfenster wird eher großzügig eingestellt, sodass eher zu viel als zu wenig Merkmale zu A_κ gehören, da eine weitere Einengung durch das Hintergrundmodell erfolgt.

Hintergrundmodell

Ein heterogener Hintergrund, über den keine Information vorliegt, wird durch eine *gleichförmige* Verteilungsdichte $p(\boldsymbol{c}_m | \boldsymbol{B}_0)$ modelliert. Eine Zuordnungsfunktion $\boldsymbol{\zeta} \in \{0,1\}^{N_{A_\kappa}}$ weist jeden Merkmalsvektor $\boldsymbol{c}_m \in A_\kappa$ entweder dem Objekt oder dem Hintergrund zu

$$\zeta_m = \begin{cases} 0 & : \quad \boldsymbol{c}_m \in \text{ Hintergrund} \\ 1 & : \quad \boldsymbol{c}_m \in \text{ Objekt} \; . \end{cases} \tag{4.9.24}$$

Damit geht das stochastische Modell über in

$$\begin{aligned} p\left(\widetilde{C}_{A_\kappa}|\boldsymbol{a}_\kappa,\boldsymbol{r},\boldsymbol{t}\right) &= \sum_{\boldsymbol{\zeta}} p\left(\widetilde{C}_{A_\kappa},\boldsymbol{\zeta}|\boldsymbol{a}_\kappa,\boldsymbol{R},\boldsymbol{t}\right) \\ &= \prod_{\boldsymbol{c}_m\in A_\kappa}\sum_{\zeta_m} p(\zeta_m)p\left(\boldsymbol{c}_m|\zeta_m,\boldsymbol{a}_\kappa,\boldsymbol{R},\boldsymbol{t}\right) \ . \end{aligned} \tag{4.9.25}$$

Man erhält $p(\zeta_m)$ aus

$$p(\zeta_m) = \begin{cases} 1 & : \ p(\boldsymbol{c}_m|\zeta_m=1,\boldsymbol{a}_\kappa,\boldsymbol{R},\boldsymbol{t}) \geq p(\boldsymbol{c}_m|\zeta_m=0,\boldsymbol{a}_0) \\ 0 & : \ p(\boldsymbol{c}_m|\zeta_m=1,\boldsymbol{a}_\kappa,\boldsymbol{R},\boldsymbol{t}) < p(\boldsymbol{c}_m|\zeta_m=0,\boldsymbol{a}_0) \end{cases} \tag{4.9.26}$$

mit $p(\boldsymbol{c}_m|\zeta_m=1,\boldsymbol{a}_\kappa,\boldsymbol{R},\boldsymbol{t})$ gleich dem stochastischen Objektmodell (4.9.20) und $p(\boldsymbol{c}_{A,l}|\zeta_m = 0,\boldsymbol{a}_0)$ gleich dem Hintergrundmodell, d. h. eine gleichförmige Dichte. Mit diesem Ansatz können Objekte der in Bild 4.9.4 gezeigten Art in unterschiedlicher Lage auch vor heterogenem Hintergrund erkannt und lokalisiert werden.

4.10 Dimensionierungsprobleme (VA.1.1.3, 13.04.2004)

Wenn der Umfang N der gegebenen Stichprobe ω sehr groß wird, lassen sich alle interessierenden Größen – z. B. Mittelwerte und Kovarianzmatrizen in (4.2.8), (4.2.9), Fehlerwahrscheinlichkeiten in (3.9.9), (3.9.10), (4.1.39), (4.1.40), Parametermatrix $\boldsymbol{A}^*$ in (4.4.19), usw. – beliebig genau schätzen. Praktisch ist der Stichprobenumfang stets endlich, und wegen des Aufwandes, der mit der Sammlung einer Stichprobe verbunden ist, wird man bestrebt sein, keine unnötig große Stichprobe zu verwenden. Die Wichtigkeit einer angemessenen Stichprobe wurde in Abschnitt 1.3 mit Postulat 1 unterstrichen. Dort wurden auch einige qualitative Aussagen gemacht. Hier werden Ansätze und Ergebnisse diskutiert, die zu quantitativen Aussagen führen. Erfahrungsgemäß ist es auch nicht so, dass man die Fehlerrate eines Klassifikationssystems durch Hinzunahme weiterer Merkmale bei konstantem Stichprobenumfang beliebig senken kann, vielmehr kann ab einer bestimmten Zahl von Merkmalen die Fehlerrate sogar wieder zunehmen. Theoretische Untersuchungen zeigen, dass dieses im Wesentlichen mit Schätzfehlern bei endlichem Stichprobenumfang zusammenhängt. Diese Probleme sind bei der Dimensionierung eines Klassifikators mit einer endlichen Stichprobe zu beachten.

Kapazität eines Klassifikators

Eine generelle Frage ist, wieviele Muster eine Stichprobe zum Training eines Klassifikators enthalten sollte. Die mit der endlichen Stichprobe ω bestimmten Parameter $\boldsymbol{a}_\lambda$ sollen ja auch die richtige Klassifikation möglichst vieler Muster aus dem Problemkreis Ω erlauben. Zusätzlich zu der qualitativen Aussage von Postulat 1 in Abschnitt 1.3 gibt es dazu ein Ergebnis, das einen quantitativen Anhaltspunkt gibt. Es stellt für den Spezialfall $k = 2$ Klassen einen Zusammenhang zwischen der Zahl N der Muster aus ω und der Zahl m der Parameter des Klassifikators her. Für zwei Klassen genügt offensichtlich *eine* Trennfunktion

$$d(\boldsymbol{c}) \;=\; \sum_{j=1}^{m} a_j \varphi_j(\boldsymbol{c}) \; . \tag{4.10.1}$$

Gegeben seien N Merkmalsvektoren oder Punkte ${}^{\varrho}\boldsymbol{c} \in \omega$ im n-dimensionalen Merkmalsraum. Die Stichprobe ω soll keine Teilmenge mit $(m+1)$ oder mehr Punkten ${}^{\varrho}\boldsymbol{c}$ enthalten, die auf einer Fläche $d(\boldsymbol{c})$ liegen. Gesucht ist nun die Zahl $D(N,m)$ der Möglichkeiten, die N Punkte mit Flächen $d(\boldsymbol{c})$, die von m Parametern abhängen, in zwei Klassen zu zerlegen. Diese Zahl ist

$$D(N,m) = \begin{cases} 2\sum\limits_{j=0}^{m} \binom{N-1}{j} & : \quad N > m \\ 2^N & : \quad N \le m \end{cases} \tag{4.10.2}$$

gilt. Die Zahl der überhaupt möglichen Zuordnungen von N Punkten zu zwei Klassen ist 2^N, die aber mit einem gegebenen Klassifikator der Form (4.10.1) i. Allg. nicht alle realisiert werden können. Die Aussage von (4.10.2) ist, dass die Zahl der realisierbaren Klassenzuordnungen nur von der Zahl m der Parameter, aber nicht von den Funktionen φ_j abhängt – allerdings werden i. Allg. bei gleichem Wert von m mit anderen Funktionen auch andere Klassenzuordnungen realisierbar sein.

Von den überhaupt möglichen Klassenzuordnungen für N Muster werde eine zufällig ausgewählt. Die Wahrscheinlichkeit P_{Nm}, dass man irgendeine der 2^N möglichen Zuordnungen

mit einem Klassifikator mit m Parametern auch realisieren kann, ist

$$P_{Nm} = \frac{D(N,m)}{2^N} = \begin{cases} 2^{1-N} \sum_{j=0}^{m} \binom{N-1}{j} & : \quad N > m \\ 1 & : \quad N \leq m \end{cases} . \tag{4.10.3}$$

Setzt man $N = l(m+1)$, $l = 1, 2, \ldots$, so gilt

$$\lim_{m\to\infty} P_{l(m+1),m} = \begin{cases} 1 & : \quad l < 2 \\ 0,5 & : \quad l = 2 \\ 0 & : \quad l > 2 \end{cases} . \tag{4.10.4}$$

Für große Werte von m (etwa ab $m \geq 30$) nähert sich also der Verlauf von P_{Nm} einer Sprungfunktion. Die Zahl

$$\boxed{N_{\mathrm{c}} = 2(m+1)} \tag{4.10.5}$$

wird auch als **Kapazität des Klassifikators** bezeichnet. Trainiert man für $k = 2$ Klassen einen Klassifikator mit m Parametern unter Verwendung von $N < N_{\mathrm{c}}$ Mustern, so kann man fast sicher sein, dass die gesuchte Klassenzuordnung realisierbar ist bzw. dass die Stichprobe separierbar ist. Wird $N > N_{\mathrm{c}}$, so kann man fast sicher sein, dass die Stichprobe nicht separierbar ist. Eine realistische Aussage über die Eigenschaften eines Klassifikators ist also i. Allg. nur zu erwarten, wenn er mit $N > N_{\mathrm{c}}$ Mustern trainiert wurde. Der Vorteil dieser Aussage ist, dass sie völlig unabhängig von speziellen Funktionen φ_j in (4.10.1) oder von Annahmen über Verteilungsdichten der Merkmale ist. Ein Maß für die Kapazität von Trennfunktionen gibt Satz 4.12, S. 361. Diese Allgemeinheit ist allerdings auch eine Schwäche, da keinerlei Bezug auf spezielle Eigenschaften eines Problemkreises genommen wird. Beispielsweise genügt bei der speziellen Struktur der Muster in Bild 3.9.1, S. 247, links bereits je ein Muster aus Ω_1 und Ω_2, um einen linearen Klassifikator so zu trainieren, dass alle Muster richtig klassifiziert werden. Die Forderung $N > N_{\mathrm{c}}$ ist daher nur als ein Anhaltspunkt zu betrachten, der durch weitere Überlegungen zu ergänzen ist.

Typ des Klassifikators und Merkmalszahl

Der erwähnte allgemeine Zusammenhang zwischen verschiedenen Einflussgrößen hat erhebliche Aufmerksamkeit gefunden. Je nach Voraussetzungen ergeben sich „optimistische“ bzw. „pessimistische“ Ergebnisse, d. h. die geschätzte Fehlerrate ist kleiner bzw. größer als die tatsächliche. Für den Fall von $k = 2$ Klassen lässt sich eine Beziehung zwischen dem erforderlichen Stichprobenumfang N_κ je Klasse ($\kappa = 1, 2$), der Zahl der Merkmale n, dem MAHALANOBIS-Abstand der Klassen (3.9.35), S. 253, und der Zuverlässigkeit der geschätzten Fehlerwahrscheinlichkeit hergestellen. Nach dem Training des Klassifikators mit N_κ Mustern je Klasse ergibt sich aus der Klassifikation einer unabhängigen Stichprobe ein Schätzwert $\widehat{p}_{\mathrm{f}}$ (s. (4.10.7)), dessen Erwartungswert $E\{\widehat{p}_{\mathrm{f}}\}$ berechnet werden kann. Mit einer sehr großen Stichprobe, $N_\kappa \to \infty$, erhält man einen Schätzwert p_∞, der bei optimaler Klassifikation mit p_B identisch ist. Das Verhältnis $\beta = E\{\widehat{p}_{\mathrm{f}}\}/p_\infty$ wird als Maß für die Zuverlässigkeit der Schätzung verwendet. Bild 4.10.1 zeigt die Abhängigkeit zwischen N_κ, $\kappa = 1, 2$, und n für den MAHALANOBIS-Abstand 5,5, da dieser große Wert zu einer kleinen Fehlerwahrscheinlichkeit (etwa 0,3%) gehört, deren zuverlässige Schätzung wiederum eine große Stichprobe erfordert. Die Kurven wurden für den quadratischen Klassifikator (Q) in (4.2.118), S. 349, und

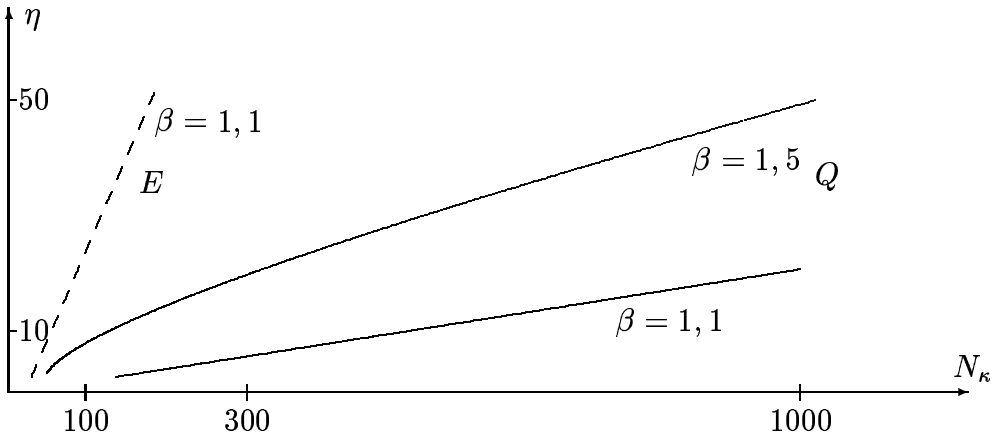


Bild 4.10.1: Zusammenhang zwischen erforderlichem Stichprobenumfang N_κ, $\kappa = 1, 2$ und sinnvoller Merkmalszahl n; Erläuterungen im Text

den EUKLID-Abstandsklassifikator (*E*) in (4.2.126), S. 350, angegeben und gelten genau für normalverteilte Merkmale mit $\mathbf{\Sigma}_\kappa = \boldsymbol{I}$ (Klassifikatoren *E, Q*), bzw. $\mathbf{\Sigma}_\kappa = \mathbf{\Sigma}_\lambda$ (Klassifikator *Q*); allerdings gelten sie näherungsweise auch für andere Verteilungen. Zum Beispiel entnimmt man Bild 4.10.1, dass bei $n = 20$ Merkmalen im Klassifikator *Q* mindestens $N_\kappa \approx 1000$ Muster zu verwenden sind, um $\beta = 1, 1$ zu erreichen, während für $\beta = 1, 5$ bereits etwa $N_\kappa \approx 300$ Muster genügen. Hat man nur $N_\kappa = 100$ Muster zur Verfügung, verwendet Klassifikator *E* und möchte $\beta = 1, 1$ erreichen, hat es keinen Sinn, mehr als etwa $n = 25$ Merkmale zu nehmen. Dieses letzte Ergebnis stimmt im Wesentlichen mit einem anderen überein, das besagt, dass für die zuverlässige Schätzung der Fehlerrate bei zwei Klassen mit normalverteilten Merkmalen und gleicher Kovarianzmatrix das Verhältnis

$$\frac{N_\kappa}{n} \approx 3 - 4 \tag{4.10.6}$$

sein sollte.

Konfidenzintervalle

Schätzwerte der Fehlerwahrscheinlichkeit wurden in (4.1.39), (4.1.40), S. 316, angegeben. Wenn man den Einfluss der Merkmalszahl und des Stichprobenumfangs auf die Schätzung der Parameter außer acht lässt, kann man Konfidenzintervalle des Schätzwertes $\widehat{p}_\mathrm{f}$ angeben. In der Stichprobe vom Umfang N werden N_f Muster falsch klassifiziert. Mit (3.9.9), S. 249, gilt

$$\widehat{p}_\mathrm{f} = \frac{N_\mathrm{f}}{N} \; . \tag{4.10.7}$$

Das Ereignis der Fehlklassifikation eines Musters tritt entweder auf oder nicht, sodass N_f binomialverteilt ist. Die Wahrscheinlichkeit für das Auftreten von N_f Ereignissen, wenn die Auftrittswahrscheinlichkeit eines Ereignisses p_f ist, erhält man aus

$$p(N_\mathrm{f}) = \binom{N}{N_\mathrm{f}} p_\mathrm{f}^N \, (1 - p_\mathrm{f})^{N - N_\mathrm{f}} \; . \tag{4.10.8}$$

Die Konfidenzintervalle dieser Dichte lassen sich numerisch recht einfach berechnen und sind in Bild 4.10.2a – Bild 4.10.2c für einige Fälle dargestellt. Wenn man z. B. den Schätzwert

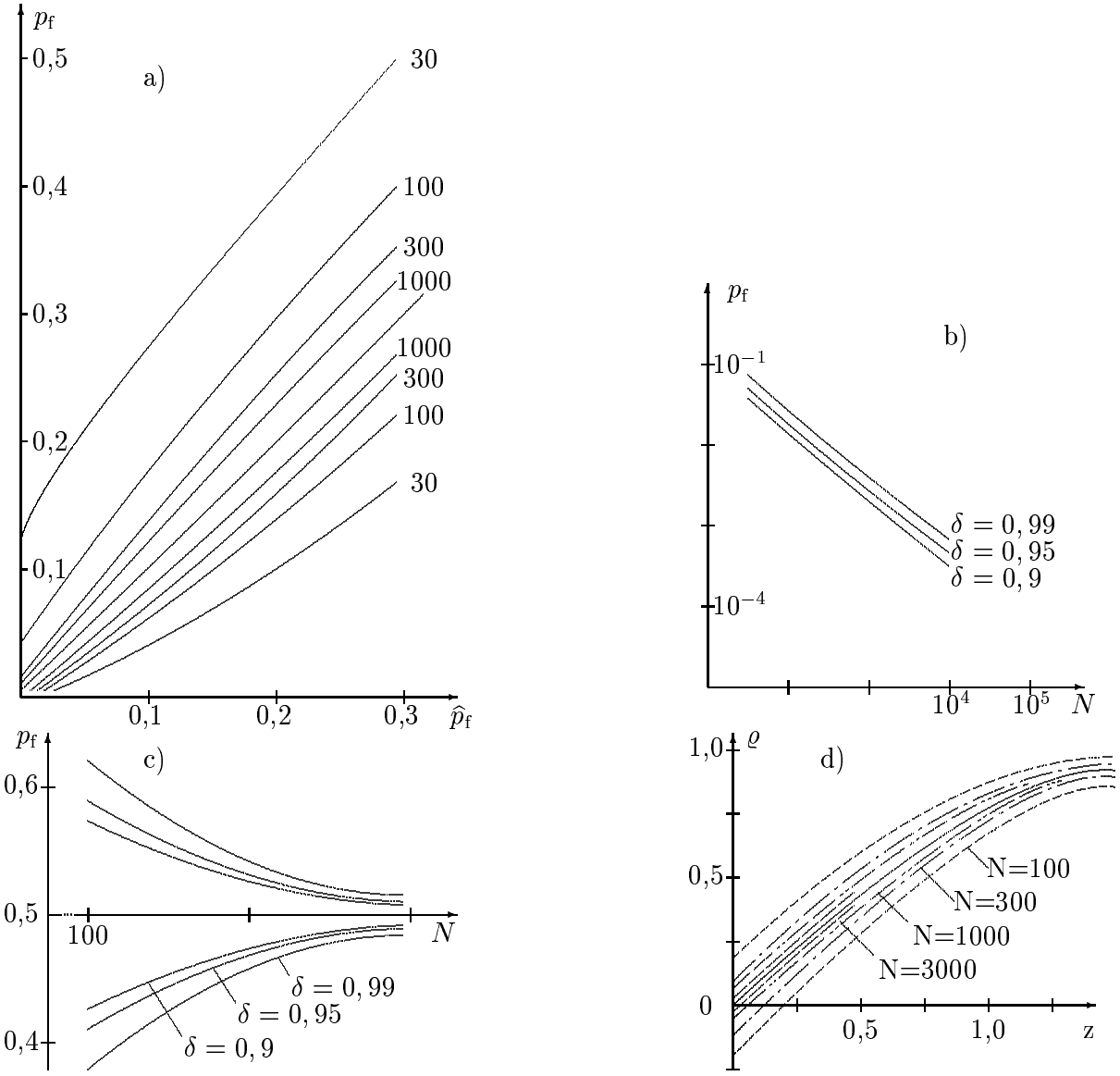


Bild 4.10.2: a) Konfidenzintervalle der Binomialverteilung auf dem Signifikanzniveau $\delta = 0,95$. b) Konfidenzintervalle für den Schätzwert $\widehat{p}_f = 0$. c) Konfidenzintervalle für den Schätzwert $\widehat{p}_f = 0,5$. d) Konfidenzintervalle des Korrelationskoeffizienten ϱ für $\delta = 0,95$, aufgetragen über $z = 0,5 \log[(1 + \widehat{\varrho})/(1 - \widehat{\varrho})]$

$\widehat{p}_f = 0,05$ berechnet, so liegt bei $N = 300$ Mustern der richtige Wert von p_f mit der Wahrscheinlichkeit $\delta = 0,95$ etwa zwischen $0,03$ und $0,08$ und bei $N = 1000$ etwa zwischen $0,04$ und $0,06$.

Bei der Dimensionierung eines Klassifikators sind häufig statistische Parameter wie Mittelwert, Streuung und Korrelationskoeffizient je Klasse zu schätzen. Die Konfidenzintervalle dieser Parameter geben Aufschluss über den Stichprobenumfang, der zu ihrer zuverlässigen Schätzung erforderlich ist. Als Maß der Zuverlässigkeit gilt die Größe eines Intervalls, in dem mit einer bestimmten Wahrscheinlichkeit (Signifikanz) der richtige Parameterwert liegt. Um einen Mittelwert oder eine Varianz mit der Signifikanz $\delta = 0,95$ auf etwa $\pm 10\%$ genau zu schätzen, sind etwa 1000 Muster je Klasse erforderlich. Korrelationskoeffizienten sind am schwierigsten

zu schätzen, daher genügt es, diese zu betrachten. In Bild 4.10.2d sind Konfidenzintervalle des Korrelationskoeffizienten ϱ über der näherungsweise normalverteilten transformierten Variablen $z = 0,5 \log[(1 + \widehat{\varrho})/(1 - \widehat{\varrho})]$ aufgetragen. Man sieht, dass mit 1000 Mustern der richtige Wert $|\varrho| \leq 0,06$ ist, wenn der Schätzwert $\widehat{\varrho} = 0$ ist.

Aufteilung der Stichprobe

In der Regel muss eine *gegebene* Stichprobe zur Schätzung der Parameter und der Fehlerwahrscheinlichkeit herangezogen werden. Dafür gibt es folgende Methoden.

1. Die Dimensionierung des Klassifikators (Lernphase) erfolgt mit der *gesamten* Stichprobe, anschließend wird mit der *gleichen* Stichprobe die Fehlerwahrscheinlichkeit geschätzt. Abgesehen von wirklich repräsentativen Stichproben sind die so ermittelten Schätzwerte optimistisch, d. h. der Schätzwert $\widehat{p}_\mathrm{f}$ ist zu klein.
2. Von der Stichprobe mit N Mustern werden $(N - 1)$ zum Training genommen, und dann wird das eine ausgelassene Muster klassifiziert, das Ergebnis notiert ("leave–one–out"). Dieses wird für alle N Muster wiederholt. Der Schätzwert ergibt sich analog aus (3.9.9) und ist besonders zuverlässig. Die Methode ist recht aufwendig, da N verschiedene Klassifikatoren zu berechnen sind. Da sich aber nur jeweils ein Muster ändert, kann die Berechnung rekursiv mit (4.2.59), (4.2.62), (4.4.36) erfolgen.
3. Als Kompromiss zwischen 1. und 2. wird die Stichprobe mit N_κ Mustern je Klasse $\omega_\kappa,\ \kappa = 1, \ldots, k$ in L disjunkte Blöcke mit je N_κ/L Mustern zerlegt, wobei N_κ/L zweckmäßigerweise ganzzahlig sein sollte. Der Klassifikator wird mit $(L - 1)$ Blöcken trainiert und mit einem Block zu je N_κ/L Mustern getestet; dieses wird L–mal mit jeweils verschiedenen Blöcken zum Test und dem Rest zum Training wiederholt. Statt N Klassifikatoren sind nur L zu realisieren. Ein Spezialfall ist $L = 2$, d. h. die Stichprobe wird in je eine gleich große Lern– und Teststichprobe zerlegt. Die für $L = 2$ gewonnenen Schätzwerte sind „pessimistisch", d. h. $\widehat{p}_\mathrm{f}$ ist zu groß.
4. In manchen Fällen, z. B. beim Training neuronaler Netze, wird neben der Trainings– und Teststichprobe noch eine Validierungsstichprobe verwendet, d. h. die gegebene Stichprobe in drei Untermengen zerlegt. Die obige Blockzerlegung kann dafür herangezogen werden. Von den L Blöcken werden $L - 2$ zum Training, einer zur Validierung und einer zum Test verwendet und das Ganze L–mal wiederholt.

4.11 Literaturhinweise

Statistische Entscheidungstheorie

Diese hat ihre Wurzeln im Hypothesentest, d. h. in der Unterscheidung zwischen den beiden Alternativen „Hypothese ist wahr“ und „Hypothese ist falsch“ sowie im Detektionsproblem, d. h. in der Unterscheidung der beiden Alternativen „Signal vorhanden“ und „kein Signal vorhanden“; Literatur dazu ist [Chernoff, 1952, Kay, 1998, Middleton, 1960, Neyman und Pearson, 1933, Wald, 1939, Wald, 1950]. Klassifikation von Mustern ist die Erweiterung auf $k > 2$ Alternativen [Abend, 1968, Berger, 1980, Chow, 1957, Devijver, 1974, Sebestyen, 1962, Therrien, 1989]. Die Möglichkeit der klassenspezifischen Klassifikation wird bereits in [VanTrees, 1968] mit der “dummy hypothesis” aufgezeigt und in [Baggenstoss, 1999, Baggenstoss, 2000, Baggenstoss, 2001] systematisch weiter entwickelt. Zur BAYES-Gleichung (4.1.3), S. 306, wird auf [Bayes, 1764, Dale, 1988] verwiesen.

Hier *nicht betrachtet* wurden Ansätze, die auf vagen Mengen basieren [Ballard und Sklansky, 1976, Bezdek, 1981, Bezdek und Pal, 1992, Borgelt und Kruse, 2003, Dubois et al., 2003, Vanderheydt et al., 1980, Zadeh et al., 1975, Zadeh, 1988], die auf Entscheidungsbäumen basieren [Chang und Pavlidis, 1977, Meisel und Michalopoulos, 1973, Payne und Meisel, 1977, Sethi und Chatterjee, 1977, Swain und Hauska, 1977], die auf syntaktischen Verfahren basieren [Fu, 1974, Fu, 1976, Fu, 1982, Gonzalez und Thomason, 1978, Kashyap, 1979, Niemann, 1974, Niemann, 1983] sowie Ansätze der Entscheidungstheorie aus den Wirtschaftswissenschaften [Bamberg und Coenenberg, 2000, Laux, 1998].

Die Kombination oder Fusion der Ergebnisse verschiedener Klassifikatoren wird bereits in [Zhuravlev, 1976, Zhuravlev, 1977a, Zhuravlev, 1977b, Zhuravlev, 1978a, Zhuravlev, 1978b, Zhuravlev und Gurevich, 1991] allgemein behandelt und theoretisch und experimentell in [Alkoot und Kittler, 1999, Barcelos et al., 2003, Ho et al., 1994, Kittler et al., 1998, Kittler und Roli, 2000, Kittler und Roli, 2001, Kuncheva, 2002, Lam und Suen, 1997, Lepistö et al., 2003, Lin et al., 2003, Murua, 2002, Schürmann, 1978, Tax et al., 2000] untersucht. Speziell für die Schrifterkennung wurde dieses in [Huang und Suen, 1995, Suen et al., 1990, Gader et al., 1996, Sirlantzkis et al., 2001, Xu et al., 1992] angewendet. Verfahren zur Generierung verschiedener Klassifikatoren werden in [Breiman, 1996, Dietterich, 2000, Freund und Shapire, 1997, Ho, 1998] behandelt.

Statistische Klassifikatoren

Die Konvergenz und Berechnung von Schätzwerten für Parameter wird in [Anderson, 1958, Kay, 1993] gezeigt. Material zu hinreichenden Statistiken geben [Frazer, 1957, Dynkin, 1961]. Die MMI- bzw. die diskriminative Schätzung wird in [Bahl et al., 1986, Dahmen et al., 1999, Ephraim et al., 1989, Katagiri et al., 1998, Kim und Un, 1988, Normandin und Morgera, 1991, Normandin et al., 1994, Normandin, 1996] behandelt, die MCE- bzw. die den Klassifikationsfehler minimierende Schätzung in [Chou et al., 1994, Ephraim und Rabiner, 1990, Juang et al., 1997, Liu et al., 1995, Purnell und Botha, 2002, Rosenberg et al., 1998, Siohan et al., 1998] und schließlich die Maximierung der Modelldistanz (MMD) in [He et al., 2000, Juang und Rabiner, 1985, Kwong et al., 1998]; diese Kriterien werden insbesondere in der Spracherkennung, teilweise auch in der Objekterkennung genutzt. Entropie Schätzungen werden in [Berger et al., 1996, Jaynes, 1982] verwendet. Zum LASSO wird auf [Osborne et al., 2000, Roth, 2004, Tibshirani, 1996] verwiesen, sparsame Schätzungen werden z. B. in

[Chen et al., 1998, Figueiredo, 2003, Tipping, 2001, Tipping, 2000, Williams, 1995] behandelt. Zur Bestimmung der Modellordnung wird auf [Akaike, 1978, Burbham und Anderson, 2002, Linhart und Zucchini, 1986, Rissanen, 1978, Stoica und Selén, 2004] verwiesen. Die Eigenschaften und die Berechnung von Schätzwerten, insbesondere auch rekursiver BS für Mittelwert und Kovarianzmatrix einer GAUSS-Verteilung, werden in [Cramér, 1974, Keehn, 1965, Niemann, 1974] behandelt; in [Keehn, 1965] wird die Eigenschaft der Selbstreproduktion von (4.2.70) gezeigt und die Integration von (4.2.75) durchgeführt. Die Parameterschätzung erfordert i. Allg. eine *große* Stichprobe; in [Martinez, 2002] wird ein probabilistischer Ansatz vorgestellt, der für die Gesichtserkennung nur *ein* Stichprobenelement erfordert.

Übersichten über unterschiedliche Aspekte der statistischen Modellierung geben [Hornegger et al., 1999, Jain et al., 2000, Silverman, 1985, Zhu, 2003]. Statistische Tests für Verteilungsannahmen werden in [Kreyszig, 1967] behandelt; Beispiele für Untersuchungen zu Konfidenzintervallen von Kovarianzmatrizen und Tests auf Normalverteilung enthält [Niemann, 1969]. Weitere experimentelle Untersuchungen zum Normalverteilungsklassifikator enthalten [Regel, 1982, Niemann, 1971, Crane et al., 1972]. Eine Dichteschätzung basierend auf einer Kombination einer nichtparametrischen und einer parametrischen Schätzung wird in [Hoti und Holmström, 2004] vorgeschlagen. Zu anderen n-dimensionalen Verteilungsdichten wird auf [Berger, 1980] verwiesen; die mehrdimensionalen Erweiterungen sind in [Cooper, 1963, Niemann, 1974] beschrieben. Sphärisch symmetrische Verteilungen werden in [Dempster, 1969] ausführlich behandelt. Elliptisch symmetrische und quadratisch asymmetrische Verteilungen werden in [Ritter et al., 1995, Ritter und Gaggenmeier, 1999] für die Chromosomenklassifikation eingesetzt. In [Srivastava et al., 2002] werden modifizierte BESSEL-Funktionen verwendet. Die Berücksichtigung statistischer Abhängigkeiten erster Ordnung in Histogrammen erfolgt in [Kazakos und Cotsidas, 1980]. Die Schätzung von Verteilungsdichten wird in [Silverman, 1986] behandelt. Eine Übersicht über statistische Ansätze zur Detektion von nicht in den Trainingsdaten enthaltenen Mustern (*Neuigkeitsdetektion*) gibt [Markou und Singh, 2003a].

Ein Ansatz zur iterativen Aufnahme weiterer Mischungskomponenten wird in [Alba et al., 1999] entwickelt. Mischungen von GAUSS-Verteilungen werden auch in [Figueiredo und Jain, 2002, Roberts et al., 1998] behandelt, wobei erstere Arbeit auch auf die Ermittlung der Zahl der Komponenten eingeht, Mischungen von Gamma-Verteilungen in [Webb, 2000]. Einige Beispiele für die Verwendung von Mischungsverteilungen zur Objekterkennung sind [Dahmen et al., 1998, Dahmen et al., 2000], zur Sprecheridentifikation und -verifikation [Reynolds und Rose, 1995, Reynolds, 1995]. Mischungsverteilungen werden in [Yang und Zwolinski, 2001] mit der Transinformation ("mutual information") geschätzt, in [Titsias und Likas, 2001] mit speziellen radialen Basisfunktionen, in [Titsias und Likas, 2003] mit gemeinsamen Komponenten. Die Verwendung von Merkmalen aus der Hauptachsentransformation (s. (3.8.34) in Abschnitt 3.8.2) und einer Mischungsverteilung wird auch als „HAT-Mischungsmodell" (bzw. "PCA mixture model") bezeichnet [Kim et al., 2003].

Eine ausführliche Darstellung von MARKOV-Zufallsfeldern gibt [Winkler, 1995]. Ein Algorithmus zur Ermittlung aller maximalen Cliquen eines Graphen ist in [Bron und Kerbosch, 1973] angegeben.

BAYES-Netzwerke werden in [Jensen, 1996, Pearl, 1988] ausführlich behandelt, Nutzungen in der Mustererkennung werden in [Berler und Shimony, 1997, Rehg et al., 2003] vorgestellt.

(Hidden) MARKOV-Modelle wurden in [Bahl et al., 1974, Bahl und Jelinek, 1975,

Bahl et al., 1983, Levinson et al., 1983] eingeführt und werden in [Rabiner, 1988, Rabiner und Juang, 1993, Schukat-Talamazzini, 1995] ausführlich behandelt. In [Bahl und Jelinek, 1975] werden die Ausgaben in Abhängigkeit von Zustandsübergängen betrachtet. So genannte Strukturierte MARKOV-Modelle werden in [Wolfertstetter und Ruske, 1995] eingeführt.

Modelle mit maximaler Entropie werden in [Jaynes, 1982] insbesondere für die spektrale Analyse erörtert; dieser Arbeit sind auch die zwei numerischen Beispiele sowie der MAXENT-Algorithmus in Abschnitt 4.2.4 entnommen. Sie gehen zurück auf [Gibbs, 1905]. Weitere Arbeiten dazu sind [Van Campenhout und Cover, 1981, Kapur und Kesavan, 1992, Bourne, 2003]. Sie werden z. B. in der maschinellen (statistischen) Übersetzung genutzt [Berger et al., 1996, Och, 2002, Och und Ney, 2002], zur Sprachmodellierung [Lau et al., 1993, Martin et al., 1990, Rosenfeld, 1996, Simons et al., 1997], Gesichtserkennung [Liu et al., 2001] oder Texturerkennung [Zhu et al., 1997, Zhu et al., 1998, Zhu et al., 2000]. Zur Auswahl der Indikatorfunktionen ("feature functions") wird auf [Della Pietra et al., 1997] verwiesen. Die Version des GIS-Algorithmus in Bild 4.2.4, S. 348, ist in [Simons et al., 1997] angegeben, die Grundlagen wurden in [Darroch und Ratcliff, 1972] gelegt und in [Della Pietra et al., 1997] erweitert.

In [Sebestyen und Edie, 1966] wird ein Beispiel für die Konstruktion n-dimensionaler Histogramme gegeben. Die Schätzung mit konstantem m_κ wurde in [Loftsgaarden und Quesenbury, 1965] eingeführt. Die PARZEN-Schätzung wird in [Parzen, 1962, Murthy, 1965, Duin, 1976] behandelt und z. B. in [Kraaijveld, 1996] zur Klassifikation genutzt. In [Breiman et al., 1977, Katkovnik und Shmulevich, 2002, Raudys, 1991, Silverman, 1978] werden Ansätze zur Bestimmung eines geeigneten Parameters h_N entwickelt. Die Verallgemeinerung der Fensterfunktion auf solche mit einer Matrix $\mathbf{\Sigma}$ statt nur eines Parameters h_N in (4.2.142) wird in [Wand und Jones, 1995] durchgeführt, die Gradienten- und Modenschätzung in [Comaniciu und Meer, 2002, Fukunaga und Hostetler, 1975], die Bestimmung einer besonders geeigneten Fensterfunktion in [Scott, 1992]. Neben der im Text erwähnten Reduktion der Stichprobe durch Vektorquantisierung gibt es in der Literatur weitere Ansätze, die entweder auf einer Dekomposition der Kernfunktion beruhen [Girolami, 2002b, Lambert et al., 1999] oder einer Reduktion der Stichprobe mit geeigneten Optimierungs- oder Clusterverfahren [Babich und Camps, 1996, Girolami und He, 2003, Holmström und Hämäläinen, 1993, Jeon und Landgrebe, 1994].

Der nächster Nachbar Klassifikator wurde in [Cover und Hart, 1967, Cover, 1969] eingeführt und ist z. B. in [Duda und Hart, 1972a, Niemann, 1974, Niemann, 1983, Devijver und Kittler, 1982] behandelt. Die Verwendung von Kernfunktionen (s. Abschnitt 3.8.3) wird in [Peng et al., 2004, Shawe-Taylor und Cristianini, 2004, Zhang et al., 2006] diskutiert. Eine Sammlung von Arbeiten enthält [Dasarathy, 2002]. Abschätzungen der Fehlerwahrscheinlichkeiten bei Rückweisungen gibt [Hellman, 1970] für das Zweiklassenproblem. Vorschläge zur Verdichtung einer Stichprobe enthalten [Hart, 1968, Gates, 1972, Gowda und Krishna, 1979, Tomek, 1976], zur Editierung [Devijer und Kittler, 1980]; weiter wird auf die obigen Bemerkungen im Zusammenhang mit der PARZEN-Schätzung verwiesen. Zur effizienten Berechnung der nächsten Nachbarn gibt es zahlreiche Vorschläge in [Cha und Srihari, 2002a, Fukunaga und Narendra, 1975, Micó et al., 1996, Moreno-Seco et al., 2003, Niemann und Goppert, 1988, Orchard, 1991, Ramasubramanian und Paliwal, 1990, Sproull, 1991, Weidong und Zheng, 1986]; ein Vergleich von schnellen Algorithmen zur Berechnung des nächsten Nachbarn findet sich in [Ramasubramanian und Paliwal, 2000]. Modifizierungen der Metrik bei kleinem Stich-

probenumfang werden in [Domeniconi et al., 2002, Peng et al., 2003] eingeführt. Verfahren bei der Suche in Multimedia Datenbanken gibt [Böhm et al., 2001], in großen Datenbeständen [Smith, 1994]. Die Anpassung der Nachbarschaft beim m-NN-Klassifikator behandelt [Wang et al., 2006].

Zum Training zahlreicher Klassifikatoren durch "stacking", "bagging" und "boosting" sowie deren Kombination wird auf [Breiman, 1996, Freund und Schapire, 1996, Friedman et al., 2000, Ho et al., 1994, Kittler et al., 1998, Kittler und Roli, 2001, Schapire, 1990, Schapire und Singer, 1999, Wolpert, 1992] verwiesen.

Support Vektor Maschinen

Eine Reihe von Arbeiten und Büchern dazu sind [Burges, 1998, Cortes und Vapnik, 1995, Cristiani und Shawe-Taylor, 2000, Schölkopf, 1997, Vapnik, 1998]; Zusammenfassungen wichtiger Ergebnisse enthalten [Müller et al., 2001, Vapnik, 1999]; zahlreiche weitere Arbeiten enthält das Sonderheft [Campbell et al., 2003]. Die Optimierung mit Nebenbedingungen, darunter auch die konvexe quadratische Optimierung, wird z. B. in Part 2, insbesondere Chap. 9, von [Fletcher, 1987] ausführlich behandelt. Die Konvergenz der Dekompositionsverfahren wird in [Lin, 2001, Lin, 2002] behandelt. Hinweise zur numerischen Lösung finden sich in [Burges, 1998, More und Wright, 1993]. Die Ersetzung der quadratischen Optimierung durch minimale Fehlerquadratlösungen wird in [Chua, 2003, Navia-Vázquez et al., 2001, Suykens und Vandewalle, 1999] beschrieben. Arbeiten zum effizienten Training sind [Chua, 2003, Dong et al., 2002, Flake und Lawrence, 2002, Platt, 1998, Zhang et al., 2006]. Der Einfluss einer Normierung der Kernfunktion wird in [Graf et al., 2003] untersucht, die Berechnung von Kernfunktionen in [Lanckriet et al., 2004]. Ein experimenteller Vergleich verschiedener Ansätze für das Mehrklassenproblem wird in [Hsu und Lin, 2002] durchgeführt; daraus gehen auch die am Anfang von Abschnitt 4.3 gemachten Bemerkungen zur Strategie „eine gegen eine" für das Mehrklassenproblem hervor sowie Ansätze zur Lösung des Mehrklassenproblems in einem Schritt. Das zur nichtlinearen Abbildung inverse Problem ("pre-image problem") wird in [Bakir et al., 2004, Kwok und Tsang, 2004, Mika et al., 1999] behandelt.

Anwendungen werden in [Pavlidis et al., 2001] behandelt. Der Zusammenhang zwischen SVM und "boosting"-Ansätzen wird in [Rätsch et al., 2002] untersucht; zum "boosting" wird auf [Breiman, 1999, Dietterich, 1999, Drucker et al., 1993, Valiant, 1984] verwiesen. Varianten der SVM sind z. B. die IVM (Import Vector Machine) [Zhu und Hastie, 2005], LSVM (Lagrangian SVM) [Mangasarian und Musicant, 2001], LS-SVM (Least Square SVM) [Suykens und Vandewalle, 1999], SSVM (Smooth SVM) [Lee und Mangasarian, 2001b], RSVM (Reduced SVM) [Lee und Mangasarian, 2001a, Lin und Lin, 2003], RVM (Relevance VM) [Tipping, 2000, Tipping, 2001, Williams et al., 2005]; weitere Ansätze geben [Chu et al., 2004, Schölkopf et al., 2000, Steinwart, 2003]. Die Konversion der SVM Ausgabe in ein probabilistisches Maß wird in [Platt, 1999] vorgeschlagen.

Ansätze wie SVM, RSVM oder RVM sind Beispiele für die Berechnung sparsamer Schätzwerte von Regressions- bzw. Klassifikationsfunktionen. Grundlagen dafür sind die Arbeiten [Breiman, 1993, Tibshirani, 1996].

Polynomklassifikatoren

Der Polynomklassifikator wird in [Meyer-Brötz und Schürmann, 1970, Schürmann, 1971, Schürmann, 1977] ausführlich behandelt; er wird auch als Quadratmittel-Klassifikator

bezeichnet. Insbesondere auch zu Abschnitt 4.4.4 finden sich in [Schürmann, 1977] zahlreiche weitere Einzelheiten. Einige verwandte Ansätze werden auch in Abschnitt 3.4 von [Niemann, 1974] erörtert. Weitere Arbeiten dazu sind [Schürmann, 1978, Schürmann und Becker, 1978, Schürmann, 1982]. Ein Vergleich mit einem statistischen Klassifikator wird in [Schürmann und Krause, 1974] durchgeführt.

Zum GAUSS-JORDAN-Algorithmus wird auf [Schürmann, 1977, Schwarz et al., 1968] verwiesen, zur Wahl der Schrittweite β_N bei der iterativen Lösung auf [Todd, 1962, Niemann und Weiss, 1979]. Stochastische Approximation wird in [Albert und Gardner, 1967, Benveniste et al., 1990, Gladyshev, 1965, Tsypkin, 1973, Wasan, 1969] behandelt. Die hier nicht behandelten iterativen Verfahren für stückweise lineare Trennfunktionen findet man in [Nilsson, 1965, Duda und Fossum, 1966, Takiyama, 1978, Takiyama, 1981].

Neuronale Netze

Die mathematische Modellierung neuronaler Aktivitäten wurde bereits in [McCulloch und Pitts, 1943] begonnen. Einschichtige Netze vom Perzeptrontyp wurden in [Nilsson, 1965, Rosenblatt, 1961] eingeführt und ihre Grenzen in [Minsky und Papert, 1969] aufgezeigt. Mit der Möglichkeit des Trainings mehrschichtiger Maschinen ergab sich eine Leistungssteigerung, die neue Perspektiven eröffnete [Chen, 1997, Cichocki und Unbehauen, 1994, Goerick et al., 1996, Hall und Bensaid, 1992, Hampshire und Waibel, 1990, Ishibuchi et al., 1993, Mao und Jain, 1995, Murino und Vernazza, 2001, Niemann und Wu, 1993, Poli et al., 1991, Wei und Hirzinger, 1994]; umfassende Bücher sind [Arbib, 1995, Cichocki und Unbehauen, 1994, Grossberg, 1988, Khanna, 1990, Kung, 1993, Michie et al., 1994, Morgan und Scofield, 1991, Ritter et al., 1991, Rumelhart und McClelland, 1986, Zell, 1994], eine einführende Übersicht ist [Lippman, 1987]. Eigenschaften von Mehrschicht-Perzeptron für die Approximation von Funktionen wurden in [Hornik et al., 1989, Hornik, 1993, Leshno et al., 1993, Makhoul et al., 1989, Muroga, 1971, White, 1990] untersucht.

Die Fehlermaße $\varepsilon_{\mathrm{MSE}}$, $\varepsilon_{\mathrm{McC}}$, $\varepsilon_{\mathrm{CE}}$, bzw. $\varepsilon_{\mathrm{CFM}}$ in Abschnitt 4.5.2 wurden in [Rumelhart und McClelland, 1988, Haffner et al., 1989, Hinton, 1990] bzw. [Hampshire und Waibel, 1990] eingeführt. Arbeiten zum Training sind [Alder et al., 1993, Baldi und Hornik, 1995, Haffner et al., 1989, Kitano, 1990, Piché, 1994, Plagianakos et al., 2002] und zur automatischen Optimierung der Struktur [Angeline et al., 1994, Braun und Weisbrod, 1993, Braun und Zagorski, 1994]. Die Komplexität neuronaler Netze, d. h. die notwendige Zahl von Neuronen, wird in [Bartlett, 1998, Huang, 2003, Tamura und Tateishi, 1997] untersucht.

Eine Übersicht über aktuelle Anwendungen neuronaler Netze in der Signalverarbeitung gibt [Chen, 1997], das Sonderheft [Murino und Vernazza, 2001] ist ihrer Anwendung in der Bildverarbeitung gewidmet. In [Markou und Singh, 2003b] wird eine Übersicht über Ansätze zur Detektion von nicht in den Trainingsdaten enthaltenen Mustern (*Neuigkeitsdetektion*) mit neuronalen Netzen gegeben. Die Verwendung von Wavelets bei der Funktionsapproximation wird in [Zhang und Benveniste, 1992] untersucht. Für nichtlineare Hauptachsentransformation mit neuronalen Netzen wird auf [Kramer, 1991, Malthouse, 1998] verwiesen.

Zu Netzen mit radialen Basisfunktionen wird auf die oben genannten Bücher zu neuronalen Netzen verwiesen, zur Optimierung aller Parameter auf [Gonzáles et al., 2003]. Netze mit zirkulären Knoten werden in [Kirby und Miranda, 1996, Ridella et al., 1997] beschrieben. Erwei-

terungen und Trainingsalgorithmen zur Wahl der Knoten und Modifikation der Gewichte geben [Alexandridis et al., 2003, Billings und Zheng, 1995, Chen et al., 1990, Chen et al., 1991, Leonardis und Bischof, 1998, Moody und Darken, 1989, Sarimveis et al., 2002]. Eigenschaften dieser Netze für die Approximation von Funktionen wurden in [Chen und Chen, 1995, Liao et al., 2003, Park und Sandberg, 1991, Park und Sandberg, 1993] untersucht.

Die Merkmalskarte wurde in [Kohonen, 1982, Kohonen, 1989, Kohonen, 1990b] vorgestellt. Eine Variante ist die lernende Vektorquantisierung [Kohonen, 1990a, Wu und Yang, 2006]. Die meisten der oben zu neuronalen Netzen zitierten Bücher behandeln sie ebenfalls. Beispiele für ihre Nutzung geben [Lakany et al., 1997, Wei und Hirzinger, 1994]. Eine hierarchische Version der Merkmalskarte ist in [Rauber et al., 2002] beschrieben. Eine verbesserte Version zur Visualisierung von Daten ist [Yin, 2002].

Andere Klassifikatortypen

Eine Grundlage für sequentielle Klassifikatoren wurde mit dem sequentiellen Wahrscheinlichkeitstest in [Wald, 1957] gelegt, weitere Literatur dazu ist [Chien und Fu, 1966, Fu, 1968, Fu und Mendel, 1970, Hussain, 1972, Hussain, 1974, Niemann, 1974].

(Binäre) Klassifikationsbäume werden in [Armstrong und Gecsei, 1979, Ballard und Sklansky, 1976, Chang und Pavlidis, 1977, Henrichon und Fu, 1969, Kulkarni und Kanal, 1976, Meisel und Michalopoulos, 1973, Payne und Meisel, 1977] behandelt, nichtbinäre in [Kulkarni und Kanal, 1976, Kuhn et al., 1994]. Beispiele für die Nutzung von Klassifikationsbäumen geben [Wang und Hirschberg, 1992, Wightman und Ostendorf, 1992, Kuhn et al., 1994]. Methoden zum Training bzw. zur automatischen Konstrukion von Klassifikationsbäumen werden in [Meisel und Michalopoulos, 1973, Breiman et al., 1984, Gelfand et al., 1994, Payne und Meisel, 1977] vorgestellt, Beispiele für hierarchische Klassifikation in [Schürmann, 1978, Regel, 1982]. Der in (4.6.7) angegebene Klassifikator wurde in [Ichino, 1979] beschrieben.

Das Beispiel für einen Klassifikator für nominale (symbolische) Merkmale ist [Stoffel, 1974] entnommen.

Zu allgemeinen Darstellungen der dynamischen Programmierung wird auf die Literatur in Abschnitt 1.9 verwiesen. Die nichtlineare Normierung mit Hilfe der dynamischen Programmierung wurde in der Sprachverarbeitung eingeführt [Itakura, 1975, Sakoe und Chiba, 1979, Rabiner und Schmidt, 1980, Myers et al., 1980]; dort finden sich auch Beispiele für Beschränkungen der Verzerrungsfunktion. Eine Erweiterung auf zweidimensionale Folgen gibt [Moore, 1979], die Handhabung großer Zustandsräume durch Verschmelzen von Zuständen behandelt [Raphael, 2001].

Klassifikation im Kontext

Eine Übersicht zur Kontextberücksichtigung enthält [Toussaint, 1978], weitere Arbeiten [Abend, 1968, Hussain, 1974, Raviv, 1967]. Der VITERBI-Algorithmus geht auf [Viterbi, 1967] zurück und wird in [Forney, 1973, Bahl und Jelinek, 1975, Neuhoff, 1975] behandelt. Ein Beispiel für die Nutzung eines Wörterbuches gibt [Doster, 1977], für die Nutzung von n-Grammen [Riseman und Hanson, 1974]. Relaxationsverfahren werden in [Eklundh et al., 1980, Niemann, 1974, Rosenfeld et al., 1976, Zucker et al., 1977] behandelt, einen Überblick über Relaxationsverfahren gibt [Kittler und Illingworth, 1985], auf die Bestimmung der Kompatibilitätskoeffizienten wird in [Pelillo und Refice, 1994, Yamamoto, 1979] eingegangen.

MARKOV-Zufallsfelder

Beispiele ihrer Nutzung in der Bildverarbeitung geben [Chellappa und Chatterjee, 1985, Chen und Kundu, 1995, Cross und Jain, 1983, Devijver, 1986, Devijver und Dekesel, 1988, Hassner und Sklansky, 1978, Kim und Yang, 1995, Laferte et al., 2000, Li, 1995, LI, 2001, Modestino und Zhang, 1992, Panjwani und Healey, 1995]; Anwendung für die Schriftzeichenerkennung [Cai und Liu, 2002].

Ein effizienter Ansatz für MARKOV-Zufallsfelder, die an Knoten eines Viererbaumes geheftet werden, wird in [Laferte et al., 2000] entwickelt. In [Storvik und Dahl, 2000] wird das Problem mit ganzzahliger linearer Programmierung gelöst. Bücher dazu sind [Li, 1995, LI, 2001, Winkler, 1995].

Texturklassifikation mit MARKOV-Feldern wird in [Chellappa und Chatterjee, 1985, Noda et al., 2002, Yang und Liu, 2002a] behandelt.

Unüberwachtes Lernen

Die Terme unüberwachtes Lernen, [Cooper und Cooper, 1964, Cooper, 1969, Patrick und Costello, 1970], Lernen ohne Lehrer, [Spragins, 1966], und Identifikation von Mischungen laufen in diesem Abschnitt praktisch auf dasselbe hinaus; die Identifizierbarkeit verschiedener Familien wird in [Teicher, 1961, Teicher, 1963] untersucht, in [Yakowitz und Spragins, 1968, Yakowitz, 1970] u. a. der Zusammenhang zu linearer Unabhängigkeit der Komponentendichten gezeigt. Weitere Arbeiten dazu sind [Alba et al., 1999, Dahmen et al., 2000, Gauvain und Lee, 1994, Martinez und Vitria, 2000, Postaire und Vasseur, 1981, Wolfe, 1967, Wolfe, 1970]. Einen Ansatz zur automatischen Ermittlung der Zahl der Mischungskomponenten bei der Identifikation von Mischungsverteilungen gibt [Yang und Liu, 2002b] (ein anderer wurde in (4.8.23) gegeben); weitere werden in [Figueiredo und Jain, 2002, Stephens, 2000] beschrieben. Die Kriterien minimale Nachrichtenlänge (minimum message length, MML), minimale Beschreibungslänge (minimum description length, MDL) und das AKAIKE-Informationskriterium, die neben anderen zur Bestimmung der Zahl der Mischungskomponenten genutzt werden, sind in [Wallace und Boulton, 1968, Rissanen, 1978, Rissanen, 1989, Whindham und Cutler, 1992] beschrieben. MAPS werden in [Cover, 1969, Hillborn und Lainiotis, 1968, Patrick und Hancock, 1966, Patrick und Costello, 1970] berechnet. Eine Kombination des EM-Algorithmus mit einem genetischen Algorithmus zur Vermeidung der Probleme mit lokalen Minima des EM-Algorithmus wurde in [Martinez und Vitria, 2000] entwickelt; andere Ansätze dafür werden in [Ueda et al., 2000, Z. et al., 2003, Zhang et al., 2004] vorgestellt. Eine Kombination der Modellierung mit Mischungsverteilungen und Analyse unabhängiger Komponenten gibt [Lee und Lewicki, 2002]. Das Prinzip des entscheidungsüberwachten Lernens wird in [Patrick et al., 1970, Scudder, 1965] erörtert. Zu den Grundlagen des EM-Algorithmus wird auf die in Kapitel 1 zitierte Literatur verwiesen. Die Konvergenz entscheidungsüberwachter Verfahren wird z. B. in [Agrawala, 1970, Imai und Shimura, 1976, Chittineni, 1980] untersucht. In [Frey und Jojic, 2003] wird ein auf Mischungsverteilungen basierendes stochastisches Modell um Transformationsinvarianz erweitert und Klassen mit dem EM-Algorithmus geschätzt. Ebenfalls basierend auf dem EM-Algorithmus wird in [Figueiredo, 2003] das Problem der möglichst guten Generalisierung beim überwachten Lernen untersucht.

Strukturerhaltende (nichtlineare) Abbildungen von hochdimensionalen Daten in die Ebene ("multidimensional scaling") werden in [Kruskal, 1964a, Kruskal, 1964b, Mao und Jain, 1995,

Niemann, 1979, Niemann, 1980, Olsen und Fukunaga, 1973, Sammon, 1969, Sammon, 1970, Shepard, 1962, Yin, 2002] untersucht; eine umfassende Darstellung gibt [Cox und Cox, 2001].

Verschiedene Ansätze zur Analyse von Häufungsgebieten ("cluster analysis") enthalten [Ball und Hall, 1967, Bezdek, 1980, Chen, 1973, Dunn, 1974, Fromm und Northouse, 1976, Koontz und Fukunaga, 1972, MacQueen, 1967, Mizoguchi und Shimura, 1976, Ney und Kuhn, 1980, Niemann, 1978, Rabiner et al., 1979, Schroeter und Bigün, 1995, Stepp und Michalski, 1986, Tarsitano, 2003, Yang und Wu, 2004], umfassende Darstellungen geben [Anderberg, 1973, Bock, 1974, Godehardt, 1990, Jain und Dubes, 1988, Niemann, 1974, Patrick, 1972, Tryon und Bailey, 1970, Tsypkin, 1973]. Ein häufig genutzter Typ von Algorithmen sind die "k- und c-means" Algorithmen [Bezdek, 1980, Clausi, 2002, Kan und Srinath, 2002, Pal und Bezdek, 1995, Pal et al., 2005, Tarsitano, 2003].

Die Gewichtung von Merkmalen wird in [Chana und Chinga, 2004, Frigui und Nasraoui, 2004, Girolami, 2002a, Yeung und Wang, 2002] vorgenommen. Ansätze, die auf vagen Mengen ("fuzzy sets", s. [Zadeh, 1965, Zadeh, 1988]) beruhen, werden in [Bezdek, 1981, Bezdek und Pal, 1992, Dunn, 1974, Hall und Bensaid, 1992, Hopper, 1999, Lim und Lee, 1990, Pal et al., 2005] beschrieben, zu ihrer Konvergenz wird auf [Bezdek, 1980] verwiesen. Auch kernbasierte Verfahren (s. Abschnitt 3.8.3) wurden vorgeschlagen [Girolami, 2002a]. Die RENYI-Entropie wurde in [Renyi, 1960] eingeführt; die Minimierung des informationstheoretisch begründeten nichtparametrischen Abstandsmasses (4.8.44) folgt [Gokcay und Principe, 2002]; das Informationspotential (4.8.39) wird in [Principe et al., 2000] genauer untersucht. Die Möglichkeit der Einbettung von Ähnlichkeitsdaten in einen Vektorraum wird in [Roth et al., 2003] gezeigt. Ein auf "boosting" basierender Ansatz wird in [Frossyniotis et al., 2004] entwickelt.

Graphentheoretische Verfahren werden in [Augustson und J. Minker, 1970, Godehardt, 1990, Koontz et al., 1976, Shapiro und Haralick, 1979, Zahn, 1971] entwickelt; zur Konstruktion des Minimalbaumes wird auf [Cheriton und Tarjan, 1976, Dijkstra, 1959, Gower und Ross, 1969, Kruskal, 1956, Prim, 1957] verwiesen. Die Konstruktion von Klassenhierarchien (und damit zusammenhängend Ultrametriken) wird ausführlich in [Bock, 1974] beschrieben. Die Entropie der Merkmale wird in [Hero et al., 2002] zur Konstruktion des Minimalbaumes verwendet.

Auch die Vektorquantisierung (s. Abschnitt 2.1.4) sowie einige neuronale Netze (s. Abschnitt 4.5.4 und [Kohonen, 1982, Hall und Bensaid, 1992]) erlauben die unüberwachte Bildung von Klassen.

Das Gebiet des „Data Mining" brachte neben traditionellen auch einige Algorithmen, die speziell auch für große Datenbestände anwendbar sind [Asharaf und Murty, 2003, Ganti et al., 1999, Han und Kambler, 2001, Kantardzic, 2003, Pujari, 2001].

Klassifikation und Lokalisation von Objekten

Eine Übersicht über die stochastische Modellierung von Objekten gibt [Hornegger et al., 1999], die Verwendung von Histogrammen wird in [Schiele und Crowley, 1996b, Schiele und Crowley, 1996a, Schiele und Crowley, 2000] untersucht. Zur Definition und Berechnung von Histogrammabständen wird auch auf [Cha und Srihari, 2002b, Hafner et al., 1995, Morovic et al., 2002, Rubner et al., 2000, Serratosa und Sanfeliu, 2006] verwiesen (in [Rubner et al., 2000] wird der „earth mover's distance" eingeführt). Die Ansätze aus Abschnitt 4.9.3 werden in [Pösl, 1999, Reinhold, 2003] genauer dargestellt, insbesondere auch die Schätzgleichungen für die unbekannten Parameter.

Die Erkennung und Lokalisierung zweidimensionaler Objekte wird in [Gonzáles-Linares et al., 2003] behandelt. Ansätze auf der Basis der HOUGH-Transformation werden in [Gonzáles-Linares et al., 2003, Ulrich et al., 2003] untersucht. Verfahren zur Erkennung von dreidimensionalen Objekten werden in [Chua und Jarvis, 1997, Correa und Shapiro, 2001, Johnson und Hebert, 1999, Reinhold, 2004, Reinhold et al., 2005, Stein und Medioni, 1992, Sun et al., 2003, Trazegnies et al., 2003, Yamany und Farag, 2002] vorgestellt. Spezielle Ansätze für die Klassifikation von Objekten sind [Borotschnig et al., 2000, Dahmen et al., 1998, Deinzer et al., 2001, Lowe, 1987, Reinhold et al., 2001, Roberts, 1965, Roth, 2001] Die effiziente Suche in großen Bilddatenbanken wird in [Chen et al., 2000] behandelt. Aufmerksamkeitssteuerung zum Finden potentiell interessanter Objekte wird in [Kalinke und von Seelen, 1996, Heidemann et al., 2003] eingesetzt.

Hidden MARKOV-Modelle (HMM) werden in [Cai und Liu, 2001, He und Kundu, 1991, Othman und Aboulnasr, 2003] zur Erkennung zweidimensionaler Objekte entwickelt. Das Wiederfinden von Bildern und Objekten ("image retrieval") wird in [Flickner et al., 1995, Han und Ma, 2002, Naphade und Huang, 2002, Swain und Ballard, 1991, Veltkamp et al., 2000, Wei, 2002] behandelt, in komprimierten Bildern in [Eftekhari-Moghadam et al., 2003].

Dimensionierungsprobleme

Das Ergebnis von (4.10.2) wird in [Nilsson, 1965] ausführlich begründet. Der Zusammenhang zwischen Zahl der Merkmale und Fehlerrate wird in [Van Campenhout, 1978, Kanal und Chandrasekaran, 1971] untersucht, der Einfluss des Stichprobenumfangs in [Foley, 1972, Jain und Ranganath, 1982, Raudys und Pikelis, 1980, Raudys und Jain, 1991]. Die in Bild 4.10.1 gezeigten Ergebnisse beruhen auf [Raudys und Pikelis, 1980], (4.10.6) auf [Foley, 1972], die Aussagen zur Aufteilung einer Stichprobe auf [Foley, 1972, Lachenbruch und Mickey, 1968, Niemann, 1969, Toussaint und Donaldson, 1970]; eine Literaturübersicht gibt [Toussaint, 1974]. Das Stichprobenproblem wird auch in [Beiden et al., 2003] wieder aufgegriffen. Die Binomialverteilung der Fehlerwahrscheinlichkeit wurde in [Highleyman, 1962] untersucht, die Ergebnisse in Bild 4.10.2 sind aus [Niemann, 1969, Niemann, 1970]; zu Konfidenzintervallen wird auf [Kreyszig, 1967] verwiesen.

Komplexitätsmaße für Klassifikatoren untersucht [Ho und Basu, 2002].

Literaturverzeichnis

[Abend, 1968] Abend, K. Compound decision procedures for unknown distributions and for dependent states of nature. In L.N. Kanal, Hg., *Pattern Recognition*, S. 204–249. Thompson, Washington D.C., 1968.

[Agrawala, 1970] Agrawala, A.K. Learning with a probabilistic teacher. *IEEE Trans. on Information Theory*, 16:373–379, 1970.

[Akaike, 1978] Akaike, H. A new look at statistical model identification. *IEEE Trans. on Automatic Control*, 19:716–723, 1978.

[Alba et al., 1999] Alba, J.L., Docío, L., Docampo, D., Márquez, O.W. Growing Gaussian mixtures network for classification applications. *Signal Processing*, 76:43–60, 1999.

[Albert und Gardner, 1967] Albert, A.E., Gardner, L.A. *Stochastic Approximation and Nonlinear Regression*. MIT Press, Cambridge, Mass., 1967.

[Alder et al., 1993] Alder, M., Lim, S.G., Hadingham, P., Attikiouzel, Y. Improving neural net convergence. *Pattern Recognition Letters*, 13:331–338, 1993.

[Alexandridis et al., 2003] Alexandridis, A., Sarimveis, H., Bafas, G. A new algorithm for online structure and parameter adaptation of RBF networks. *Neural Networks*, 16:1003–1017, 2003.

[Alkoot und Kittler, 1999] Alkoot, F.M., Kittler, J. Experimental evaluation of expert fusion strategies. *Pattern Recognition Letters*, 20:1361–1369, 1999.

[Anderberg, 1973] Anderberg, M.R. *Cluster Analysis for Applications*. Academic Press, New York, 1973.

[Anderson, 1958] Anderson, T.W. *Introduction to Multivariate Statistical Analysis*, Kap. 3. J. Wiley, New York, 1958.

[Angeline et al., 1994] Angeline, P.J., Saunders, G.M., Pollack, J.B. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans. on Neural Networks*, 5:54–65, 1994.

[Arbib, 1995] Arbib, M.A., Hg. *The Handbook of Brain-Theory and Neural Networks*. The MIT Press, Cambridge, Massachusetts, USA, 1995.

[Armstrong und Gecsei, 1979] Armstrong, W.A., Gecsei, J. Adaption algorithms for binary tree networks. *IEEE Trans. on Systems, Man, and Cybernetics*, 9:276–285, 1979.

[Asharaf und Murty, 2003] Asharaf, S., Murty, M.N. An adaptive rough fuzzy single pass algorithm for clustering large data sets. *Pattern Recognition*, 36:3015–3018, 2003.

[Augustson und J. Minker, 1970] Augustson, J.G., J. Minker, J. An analysis of some graph theoretical cluster techniques. *Journal of the Association for Comp. Machinery*, 17:571–588, 1970.

[Babich und Camps, 1996] Babich, G.A., Camps, O. Weighted Parzen widows for pattern classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:567–570, 1996.

[Baggenstoss, 1999] Baggenstoss, P.M. Class-specific features in classification. *IEEE Trans. on Signal Processing*, S. 3428–3432, 1999.

[Baggenstoss, 2000] Baggenstoss, P.M. A theoretically optimum approach to classification using class-specific features. In A. Sanfeliu, J.J. Villanueva, M. Vanrell, R. Alquézar, J. Crowley, Y. Shirai, Hg., *Proc. Int. Conference on Pattern Recognition (ICPR)*. (IEEE Computer Society Press), Barcelona, Spain, 2000.

[Baggenstoss, 2001] Baggenstoss, P.M. A modified Baum-Welch algorithm for hidden Markov models.

IEEE Trans. on Speech and Audio Processing, S. 411–416, 2001.

[Bahl et al., 1986] Bahl, L.R., Brown, P.F., de Souza, P.V., Mercer, R.L. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 49–52. Tokyo, Japan, 1986.

[Bahl et al., 1974] Bahl, L.R., Cocke, J., Jelinek, F., Raviv, J. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. on Information Theory*, 20:284–287, 1974.

[Bahl und Jelinek, 1975] Bahl, L.R., Jelinek, F. Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition. *IEEE Trans. on Information Theory*, 21:404–411, 1975.

[Bahl et al., 1983] Bahl, L.R., Jelinek, F., Mercer, L.R. A maximum likelihood approach to continuous speech recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5:179–190, 1983.

[Bakir et al., 2004] Bakir, G.H., Weston, J., Schölkopf, B. Learning to find pre-images. In S. Thrun, L. Saul, B. Schölkopf, Hg., *Advances in Neural Information Processing Systems*, S. 449–456. MIT Press, Cambridge, MA, USA, 2004.

[Baldi und Hornik, 1995] Baldi, P.F., Hornik, K. Learning in linear neural networks: A Survey. *IEEE Trans. on Neural Networks*, 6:837–858, 1995.

[Ball und Hall, 1967] Ball, G.H., Hall, J.D. A clustering technique for summarizing multivariate data. *Behavioral Sciences*, 12:153–155, 1967.

[Ballard und Sklansky, 1976] Ballard, D.H., Sklansky, J. A ladder structured decision tree for recognizing tumors in chest radiographs. *IEEE Trans. on Computers*, 25:503–513, 1976.

[Bamberg und Coenenberg, 2000] Bamberg, G., Coenenberg, A.G. *Betriebswirtschaftliche Entscheidungslehre*. Vahlen, München, Germany, 10. Aufl., 2000.

[Barcelos et al., 2003] Barcelos, C.A.Z., Boaventura, M., Silva, E.C. A well-balanced flow equation for noise removal and edge detection. *IEEE Trans. on Image Processing*, 14:751–763, 2003.

[Bartlett, 1998] Bartlett, P.L. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. on Information Theory*, 44:525–536, 1998.

[Bayes, 1764] Bayes, T. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. Roy. Soc.*, S. 370–418, 1764.

[Beiden et al., 2003] Beiden, S.V., Maloof, M.A., Wagner, R.F. A general model for finite-sample effects in training and testing of competing classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1561–1569, 2003.

[Benveniste et al., 1990] Benveniste, A., Métivier, M., Priouret, P. *Adaptive algorithms and stochastic approximations*, Bd. 22 von *Applications of mathematics*. Springer, Berlin, 1990.

[Berger et al., 1996] Berger, A.L., Della Pietra, S.A., Della Pietra, V.J. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, 1996.

[Berger, 1980] Berger, J.O. *Statistical Decision Theory, Foundations, Concepts, and Methods*. Springer, Berlin, Heidelberg, 1980.

[Berler und Shimony, 1997] Berler, A., Shimony, S.E. Bayes networks for sonar sensor fusion. In *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1997.

[Bezdek und Pal, 1992] Bezdek, C.J., Pal, S.K., Hg. *Fuzzy Models for Pattern Recognition*. Inst. of Electrical and Electronic Engineers, IEEE Press, New York, 1992.

[Bezdek, 1980] Bezdek, J.C. A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:1–8, 1980.

[Bezdek, 1981] Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[Billings und Zheng, 1995] Billings, S.A., Zheng, G.L. Radial basis function network configuration using genetic algorithms. *Neural Networks*, 8:877–890, 1995.

[Bock, 1974] Bock, H.H. *Automatische Klassifikation*. Vandenhoeck und Ruprecht, Göttingen, 1974.

[Böhm et al., 2001] Böhm, C., Berchtold, S., Keim, D.A. Searching in high-dimensional spaces – index

structures for improving performance of multimedia databases. *ACM Computing Surveys*, 33:322–373, 2001.

[Borgelt und Kruse, 2003] Borgelt, C., Kruse, R. Operations and evaluation measures for learning possibilistic graphical models. *Artificial Intelligence*, 148(1-2):385–418, 2003.

[Borotschnig et al., 2000] Borotschnig, H., Paletta, L., Prantl, M., Pinz, A. Appearance-based active object recognition. *Image and Vision Computing*, 18:715–727, 2000.

[Bourne, 2003] Bourne, R.A. Explaining default intuitions using maximum entropy. *Journal of Applied Logic*, 1:255–271, 2003.

[Braun und Weisbrod, 1993] Braun, H., Weisbrod, J. Evolving neural networks for application oriented problems. In *Proc. of the Second Annual Conference on Evolutionary Programming*, S. 62–71. Evolutionary Programming Society, San Diego, California, USA, 1993.

[Braun und Zagorski, 1994] Braun, H., Zagorski, P. ENZO-M – A hybrid approach for optimizing neural networks by evolution and learning. In *Proc. 3. Conference on Parallel Problem Solving from Nature*, S. 440–451. Springer, LNCS 866, Berlin, Heidelberg, 1994.

[Breiman, 1993] Breiman, L. Better subset selection using the non-negative garotte. Techn. Ber., University of California, Berkeley, CA, 1993.

[Breiman, 1996] Breiman, L. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Breiman, 1999] Breiman, L. Prediction games and arcing algorithms. *Neural Computation*, 11:1493–1518, 1999.

[Breiman et al., 1984] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. *Classification and Regression Trees*. Wadsworth Int., Belmont, CA, 1984.

[Breiman et al., 1977] Breiman, L., Meisel, W., Purcell, E. Variable kernel estimates of multivariate densities. *Technometrics*, 19:261–270, 1977.

[Bron und Kerbosch, 1973] Bron, C., Kerbosch, J. Finding all cliques of an undirected graph. *Communic. of the Association for Computing Machinery*, 16:575–577, 1973.

[Burbham und Anderson, 2002] Burbham, K.P., Anderson, D.R. *Model Selection and Multi-Model Inference*. Springer, Berlin Heidelberg, 2002.

[Burges, 1998] Burges, C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[Cai und Liu, 2001] Cai, J., Liu, Z.-Q. Hidden Markov models with spectral features for 2D shape recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:1454–1458, 2001.

[Cai und Liu, 2002] Cai, J., Liu, Z.-Q. Pattern recognition using Markov random field models. *Pattern Recognition*, 35:725–733, 2002.

[Campbell et al., 2003] Campbell, C., Lin, C.-J., Keerthi, S.S., Sanchez, V.D. Editorial: Special issue on support vector machines. *Neurocomputing*, 55(1-2):1–3, 2003.

[Cha und Srihari, 2002a] Cha, S.-H., Srihari, S.N. A fast nearest neighbor search algorithm by filtration. *Pattern Recognition*, 35:515–525, 2002a.

[Cha und Srihari, 2002b] Cha, S.-H., Srihari, S.N. On measuring the distance between histograms. *Pattern Recognition*, 35:1355–1370, 2002b.

[Chana und Chinga, 2004] Chana, E.Y., Chinga, W.K. An optimization algorithm for clusterung using weighted dissimilarity measures. *Pattern Recognition*, 37:943–952, 2004.

[Chang und Pavlidis, 1977] Chang, R.L.P., Pavlidis, T. Fuzzy decision tree algorithms. *IEEE Trans. on Systems, Man, and Cybernetics*, 7:28–34, 1977.

[Chellappa und Chatterjee, 1985] Chellappa, R., Chatterjee, S. Classification of textures using Gaussian Markov random fields. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 33:559–563, 1985.

[Chen, 1973] Chen, C.H. *Statistical Pattern Recognition*. Hayden, New York, 1973.

[Chen et al., 2000] Chen, J.-Y., Bouman, C.A., Dalton, J.C. Hierarchical browsing and search of large image databases. *IEEE Trans. on Image Processing*, 9:442–455, 2000.

[Chen und Kundu, 1995] Chen, J.L., Kundu, A. Unsupervised texture segmentation using multichannel

decomposition and hidden Markov models. *IEEE Trans. on Image Processing*, 4:603–619, 1995.

[Chen et al., 1990] Chen, S., Billings, S.A., Cowan, C.F.N., Grant, P.W. Practical identification of NARMAX models using radial basis functions. *Int. Journal of Control*, 52:1327–1350, 1990.

[Chen et al., 1991] Chen, S., Cowan, C.F.N., Grant, P.M. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. on Neural Networks*, 2:302–309, 1991.

[Chen et al., 1998] Chen, S., Donoho, D., Saunders, M. Atomic decomposition by basis pursuit. *SIAM J. Scientific Computation*, 20(1):33–61, 1998.

[Chen, 1997] Chen, T. The past, present, and future of neural networks for signal processing. *IEEE Signal Processing Magazine*, 14(6):28–48, 1997.

[Chen und Chen, 1995] Chen, T., Chen, H. Universal approximation of nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. on Neural Networks*, 6:911–917, 1995.

[Cheriton und Tarjan, 1976] Cheriton, D., Tarjan, R.E. Finding minimum spanning trees. *SIAM J. of Computing*, 5:724–741, 1976.

[Chernoff, 1952] Chernoff, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[Chien und Fu, 1966] Chien, Y.T., Fu, K.S. A modified sequential recognition machine using time-varying stopping boundaries. *IEEE Trans. on Information Theory*, 12:206–214, 1966.

[Chittineni, 1980] Chittineni, C.B. Learning with imperfectly labeled patterns. *Pattern Recognition*, 12:281–291, 1980.

[Chou et al., 1994] Chou, W., Lee, C.H., Juang, B.H., Soong, F.K. A minimum error rate pattern recognition approach to speech recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 8:5–31, 1994.

[Chow, 1957] Chow, C.K. An optimum character recognition system using decision functions. *IRE Trans. Electronic Computers*, 6:247–254, 1957.

[Chu et al., 2004] Chu, W., Keerthi, S.S., Ong, C.J. Bayesian support vector regression using a unified loss function. *IEEE Trans. on Neural Networks*, 15:29–44, 2004.

[Chua und Jarvis, 1997] Chua, C.S., Jarvis, R. Point signatures: A new representation for 3-D object recognition. *Int. Journal of Computer Vision*, 25:63–85, 1997.

[Chua, 2003] Chua, K.S. Efficient computations for large least square support vector machine classifiers. *Pattern Recognition Letters*, 24:75–80, 2003.

[Cichocki und Unbehauen, 1994] Cichocki, A., Unbehauen, R. *Neural Networks for Optimization and Signal Processing*. J. Wiley, New York, 1994.

[Clausi, 2002] Clausi, D.A. K-means iterative Fisher (KIF) unsupervised clustering algorithm applied to image texture segmentation. *Pattern Recognition*, 35:1959–1972, 2002.

[Comaniciu und Meer, 2002] Comaniciu, D., Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[Cooper, 1963] Cooper, D.B. Multivariate extension of onedimensional probability distributions. *IEEE Trans. on Electronic Computers*, 12:572–573, 1963.

[Cooper und Cooper, 1964] Cooper, D.B., Cooper, P.W. Nonsupervised adaptive signal detection and pattern recognition. *Information and Control*, 7:416–444, 1964.

[Cooper, 1969] Cooper, P.W. Nonsupervised learning in statistical pattern recognition. In S. Watanabe, Hg., *Methodologies of Pattern Recognition*, S. 97–109. Academic Press, New York, 1969.

[Correa und Shapiro, 2001] Correa, S., Shapiro, L. A new signature based method for efficient 3-D object recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, S. 769–776, 2001.

[Cortes und Vapnik, 1995] Cortes, V., Vapnik, V.N. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[Cover, 1969] Cover, T.M. Learning in pattern recognition. In S. Watanabe, Hg., *Methodologies of Pattern Recognition*, S. 111–132. Academic Press, New York, 1969.

[Cover und Hart, 1967] Cover, T.M., Hart, P.E. Nearest neighbor pattern classification. *IEEE Trans. on Information Theory*, 13:21–27, 1967.

[Cox und Cox, 2001] Cox, T.F., Cox, M.A.A. *Multidimensional Scaling*. Monographs on Statistics and Applied Probability 88. Chapman & Hall, London, UK, 2. Aufl., 2001.

[Cramér, 1974] Cramér, H. *Mathematical Methods of Statistics*. Princeton Mathematical Series. Princeton University Press, Princeton, USA, 13. Aufl., 1974.

[Crane et al., 1972] Crane, R.B., Malila, W.A., Richardson, W. Suitability of the normal density assumption for processing multispectral scanner data. *IEEE Trans. on Geoscience Electronics*, 10:158–165, 1972.

[Cristiani und Shawe-Taylor, 2000] Cristiani, N., Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, England, 2000.

[Cross und Jain, 1983] Cross, G.R., Jain, A.K. Markov random field texture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5:25–39, 1983.

[Dahmen et al., 1998] Dahmen, J., Beulen, K., Ney, H. Objektklassifikation mit Mischverteilungen. In P. Levi, R.-J. Ahlers, F. May, M. Schanz, Hg., *DAGM Symposium Mustererkennung*, S. 167–174. (Springer, Berlin Heidelberg), Stuttgart, 1998.

[Dahmen et al., 2000] Dahmen, J., Hektor, J., Perrey, R., Ney, H. Automatic classification of red blood cells using Gaussian mixture densities. In A. Horsch, T. Lehmann, Hg., *Bildverarbeitung für die Medizin*, S. 331–335. (Springer, Berlin), München, 2000.

[Dahmen et al., 1999] Dahmen, J., Schlüter, R., Ney, H. Discriminative training auf Gaussian mixture densities for image object recognition. In W. Förstner, J.M. Buhmann, A. Faber, P. Faber, Hg., *Mustererkennung 1999*, S. 205–212. Springer (Berlin, Heidelberg), 21. DAGM Symposium, Bonn, 1999.

[Dale, 1988] Dale, A.I. On Bayes' theorem and the inverse Bernoulli theorem. *Historia Math.*, 15(4):348–360, 1988.

[Darroch und Ratcliff, 1972] Darroch, J.N., Ratcliff, D. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480, 1972.

[Dasarathy, 2002] Dasarathy, B.V. *Nearest Neighbor Pattern Classification Techniques*. IEEE Computer Society Press, New York, 2002.

[Deinzer et al., 2001] Deinzer, F., Denzler, J., Niemann, H. On fusion of multiple views for active object recognition. In B. Radig, S. Florczyk, Hg., *Pattern Recognition. Proc. 23rd DAGM Symposium*, S. 239–245. (Springer LNCS 2191, Berling Heidelberg, ISBN 3-540-42596-9), München, Germany, 2001.

[Della Pietra et al., 1997] Della Pietra, S., Della Pietra, V., Lafferty, J. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19:380–393, 1997.

[Dempster, 1969] Dempster, A.P. *Elements of Continuous Multivariate Analysis*. Addison-Wesley, Reading, Massachusetts, USA, 1969.

[Denzler, 1992] Denzler, J. Transformation von Sprachsignalen in Laryngosignale mittels künstlicher neuronaler Netze. Diplomarbeit, Lehrstuhl für Mustererkennung (Informatik 5), Univ. Erlangen-Nürnberg, Erlangen, 1992.

[Devijer und Kittler, 1980] Devijer, P.A., Kittler, J. On the edited nearest neighbour rule. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. 72–80. Miami, Florida, 1980.

[Devijver, 1974] Devijver, P.A. On a new class of bounds on Bayes risk in multihypothesis pattern recognition. *IEEE Trans. on Computers*, 23:70–80, 1974.

[Devijver, 1986] Devijver, P.A. Probabilistic labeling in a hidden second order Markov mesh. In E.S. Gelsema, L.N. Kanal, Hg., *Pattern Regcognition in Practice II*, S. 113–123. North Holland, Amsterdam, The Netherlands, 1986.

[Devijver und Dekesel, 1988] Devijver, P.A., Dekesel, M.M. Real-time restoration and segmentation algorithms for hidden Markov random field models. In A.K. Jain, Hg., *Real-Time Object Measurement and Classification*, Bd. 42 von *NATO ASI Series F*, S. 293–307. Springer, Berlin, Germany,

1988.

[Devijver und Kittler, 1982] Devijver, P.A., Kittler, J. *Pattern Recognition – a Statistical Approach*. Prentice Hall, Englewood Cliffs, NJ, USA, 1982.

[Dietterich, 1999] Dietterich, T.G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2), 1999.

[Dietterich, 2000] Dietterich, T.G. Ensemble methods in machine learning. In J. Kittler, F. Roli, Hg., *Proc. 1st International Workshop on Multiple Classifier Systems*, S. 1–15. (Springer, LNCS 1857), 2000.

[Dijkstra, 1959] Dijkstra, E.W. A note on two problems in connection with graphs. *Nmerical Mathematics*, 1(5):269–271, 1959.

[Domeniconi et al., 2002] Domeniconi, C., Peng, J., Gunopulos, D. Locally adaptive metric nearest-neighbor classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1281–1285, 2002.

[Dong et al., 2002] Dong, J.-X., Krzyzak, A., Suen, C.Y. A practical SMO algorithm. In R. Kasturi, D. Laurendeau, C. Suen, Hg., *Proc. Int. Conference on Pattern Recognition (ICPR)*. InControl Productions Inc., Monterey, CA (ISBN 0-7695-1699-8), Quebec City, Canada, 2002.

[Doster, 1977] Doster, W. Contextual postprocessing system for cooperation with a multiple-choice character-recognition system. *IEEE Trans. on Computers*, 26:1090–1101, 1977.

[Drucker et al., 1993] Drucker, H., Shapire, R., Simard, P.Y. Boosting performance in neural networks. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 7:705–719, 1993.

[Dubois et al., 2003] Dubois, D., Fargier, H., Perny, P. Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach,. *Artificial Intelligence*, 148(1-2):219–260, 2003.

[Duda und Fossum, 1966] Duda, R.O., Fossum, H. Pattern classification by iteratively determined linear and piecewise linear discriminant functions. *IEEE Trans. on Electronic Computers*, 15:220–232, 1966.

[Duda und Hart, 1972a] Duda, R.O., Hart, P.E. *Pattern Classification and Scene Analysis*. J. Wiley, New York, 1972a.

[Duda und Hart, 1972b] Duda, R.O., Hart, P.E. Use of Hough transformation to detect lines and curves in pictures. *Communic. of the Association for Computing Machinery*, 15:11–15, 1972b.

[Duin, 1976] Duin, R.P.W. On the choice of smoothing parameters for Parzen estimators of probability density functions. *IEEE Trans. on Computers*, 25:1175–1179, 1976.

[Dunn, 1974] Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.*, 3(3):32–57, 1974.

[Dynkin, 1961] Dynkin, E.B. Necessary and sufficient statistics for a class of probability distributions. *Selected Transl. in Math. Statistics and Probability*, 1:17–40, 1961.

[Eftekhari-Moghadam et al., 2003] Eftekhari-Moghadam, A.M., Shanbehzadeh, J., Mahmoudi, F., Soltaninan-Zadeh, H. Image retrieval based on index compressed vector quantization. *Pattern Recognition*, 36:2635–2647, 2003.

[Eklundh et al., 1980] Eklundh, J.O., Yamamoto, H., Rosenfeld, A. A relaxation method for multispectral pixel classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:72–75, 1980.

[Ephraim et al., 1989] Ephraim, Y., Dembo, A., Rabiner, L.R. A minimum discrimination information approach for hidden Markov modeling. *IEEE Trans. on Information Theory*, 35:1001–1003, 1989.

[Ephraim und Rabiner, 1990] Ephraim, Y., Rabiner, L.R. On the relation between modeling approaches for speech recognition. *IEEE Trans. on Information Theory*, 36:372–380, 1990.

[Figueiredo, 2003] Figueiredo, M.A.T. Adaptive sparseness for supervised learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1150–1159, 2003.

[Figueiredo und Jain, 2002] Figueiredo, M.A.T., Jain, A.K. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[Flake und Lawrence, 2002] Flake, G.W., Lawrence, S. Efficient svm regression training with smo. *Machine Learning*, 46(1-3):271–290, 2002.

[Fletcher, 1987] Fletcher, R. *Practical Methods of Optimization*. J. Wiley, Chichester, 2. Aufl., 1987.

[Flickner et al., 1995] Flickner, M., Swahney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P. Querying by image and video content: The QBIC System. *IEEE Trans. on Computers*, 28:23–32, 1995.

[Foley, 1972] Foley, D.H. Considerations of sample and feature size. *IEEE Trans. on Information Theory*, 18:618–626, 1972.

[Forney, 1973] Forney, G.D. The Viterbi algorithm. *Proc. IEEE*, 61:268–278, 1973.

[Frazer, 1957] Frazer, D.A.S. *Nonparametric Methods in Statistics*. J. Wiley, New York, 1957.

[Freund und Schapire, 1996] Freund, Y., Schapire, R.E. Experiments with a new boosting algorithm. In *Proc. 13th Int. Conf. Machine Learning*, S. 148–156, 1996.

[Freund und Shapire, 1997] Freund, Y., Shapire, R.E. A decision theoretic generalization of online learning and application to boosting. *J. on Computer and Systems Sciences*, 55:119–139, 1997.

[Frey und Jojic, 2003] Frey, B.J., Jojic, N. Transformation-invariant clustering using the EM-algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1–17, 2003.

[Friedman et al., 2000] Friedman, J., Hastie, T., Tibshirani, R. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, 2000.

[Frigui und Nasraoui, 2004] Frigui, H., Nasraoui, O. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37:567–581, 2004.

[Fromm und Northouse, 1976] Fromm, F.R., Northouse, R.A. CLASS, a nonparametric clustering algorithm. *Pattern Recognition*, 8:107–114, 1976.

[Frossyniotis et al., 2004] Frossyniotis, D., Likas, A., Stafylopatis, A. A clustering method based on boosting. *Pattern Recognition Letters*, 25:641–654, 2004.

[Fu, 1968] Fu, K.S. *Sequential Methods in Pattern Recognition and Machine Learning*. Academic Press, New York, 1968.

[Fu, 1974] Fu, K.S. *Syntactic Methods in Pattern Recognition*. Academic Press, New York, 1974.

[Fu, 1976] Fu, K.S. Tree languages and syntactic pattern recognition. In C.H. Chen, Hg., *Pattern Recognition and Artifical Intelligence*, S. 257–291. Academic Press, New York, 1976.

[Fu, 1982] Fu, K.S. *Syntactic Pattern Recognition and Applications*. Prentice Hall, Englewood Cliffs, N.J., 1982.

[Fu und Mendel, 1970] Fu, K.S., Mendel, J.M. *Adaptive, Learning, and Pattern Recognition Systems*. Academic Press, New York, 1970.

[Fukunaga und Hostetler, 1975] Fukunaga, K., Hostetler, L.D. The estimation of the gradient of a density function, with applications to pattern recognition. *IEEE Trans. on Information Theory*, 21:32–40, 1975.

[Fukunaga und Narendra, 1975] Fukunaga, K., Narendra, P.M. A branch and bound algorithm for computing k-nearest neighbours. *IEEE Trans. on Computers*, 24:750–753, 1975.

[Gader et al., 1996] Gader, P.D., Mohamed, M.A., Keller, J.M. Fusion of handwritten word classifiers. *Pattern Recognition Letters*, 17:577–584, 1996.

[Ganti et al., 1999] Ganti, V., Gehrke, J., Ramakrishnan, R. Mining very large databases. *IEEE Computer*, 32(8):38–45, 1999.

[Gates, 1972] Gates, G.W. The reduced nearest neighbour rule. *IEEE Trans. on Information Theory*, 18:431–433, 1972.

[Gauvain und Lee, 1994] Gauvain, J.-L., Lee, C.-H. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. on Speech and Audio Processing*, 2:291–298, 1994.

[Gelfand et al., 1994] Gelfand, S., Ravishankar, C., Delp, E. An iterative growing and pruning algorithm for classification tree design. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:302–320, 1994.

[Gibbs, 1905] Gibbs, J.W. *Elementare Grundlagen der statistischen Mechanik*. Barth, Leipzig, 1905.

[Girolami, 2002a] Girolami, M. Mercer kernel-based clustering in feature space. *IEEE Trans. on Neural Networks*, 13:780–784, 2002a.

[Girolami, 2002b] Girolami, M. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14:669–688, 2002b.

[Girolami und He, 2003] Girolami, M., He, C. Probability density estimation from optimally condensed data samples. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36:1253–1264, 2003.

[Gladyshev, 1965] Gladyshev, E.G. On stochastic approximation. *Automatika e Telemekanika*, 10(2):275–278, 1965.

[Godehardt, 1990] Godehardt, E. *Graphs as Structural Models*, Bd. 4 von *Advances in System Analysis*. Vieweg, Braunschweig, Germany, 2. Aufl., 1990.

[Goerick et al., 1996] Goerick, C., Noll, D., Werner, M. Artificial neural networks in real time car detection and tracking applications. *Pattern Recognition Letters*, 17:335–343, 1996.

[Gokcay und Principe, 2002] Gokcay, E., Principe, J.C. Information theoretic clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:158–171, 2002.

[Gonzáles et al., 2003] Gonzáles, J., Rojas, I., Ortega, J., Pomares, H., Fernández, F.J., Díaz, A.F. Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Trans. on Neural Networks*, 14:1478–1495, 2003.

[Gonzáles-Linares et al., 2003] Gonzáles-Linares, J.M., Guil, N., Zapata, E.L. An efficient 2D deformable objects detection and location algorithm. *Pattern Recognition*, 36:2543–2556, 2003.

[Gonzalez und Thomason, 1978] Gonzalez, R.C., Thomason, M.G. *Syntactic Pattern Recognition, an Introduction*. Addison-Wesley, Reading, Mass., 1978.

[Gowda und Krishna, 1979] Gowda, K.C., Krishna, G. The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. *IEEE Trans. on Information Theory*, 25:488–490, 1979.

[Gower und Ross, 1969] Gower, J.C., Ross, G.J.S. Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, 18(1):54–64, 1969.

[Graf et al., 2003] Graf, A.B.A., Smola, A.J., Borer, S. Classification in normalized feature space using support vector machines. *IEEE Trans. on Neural Networks*, 14:597–605, 2003.

[Grossberg, 1988] Grossberg, S., Hg. *Neural Networks and Natural Intelligence*. MIT Press, Cambridge, Mass., 1988.

[Haffner et al., 1989] Haffner, P., Waibel, A., Sawai, H., Shikano, K. Fast back-propagation learning methods for large phonemic neural networks. In *Proc. European Conference on Speech Communication and Technology*, S. 553–556. Paris, 1989.

[Hafner et al., 1995] Hafner, J., Sawhney, J.S., Equitz, W., Flicker, M., Niblack, W. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:729–735, 1995.

[Hall und Bensaid, 1992] Hall, L.O., Bensaid, A.M. A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE Trans. on Neural Networks*, 3(5):672–682, 1992.

[Hampshire und Waibel, 1990] Hampshire, J.B., Waibel, A.H. A novel objective function for improved phoneme recognition using time-delay neural networks. *IEEE Trans. on Neural Networks*, 1:216–228, 1990.

[Han und Kambler, 2001] Han, J., Kambler, M. *Data Mining: Concept and Techniques*. Morgan Kaufmann, San Mateo, 2001.

[Han und Ma, 2002] Han, J., Ma, K.-K. Fuzzy color histogram and its use in color image retrieval. *IEEE Trans. on Image Processing*, 11:944–952, 2002.

[Hart, 1968] Hart, P.E. The condensed nearest neighbor rule. *IEEE Trans. on Information Theory*, 14:515–516, 1968.

[Hassner und Sklansky, 1978] Hassner, M., Sklansky, J. Markov random field models of digitized image

texture. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. 538–540. Kyoto, Japan, 1978.

[He et al., 2000] He, Q.H., Kwong, S., Man, K.F., Tang, K.S. Improved maximum model distance for HMM training. *Pattern Recognition*, 33:1749–1758, 2000.

[He und Kundu, 1991] He, Y., Kundu, A. 2-D shape classification using hidden Markov model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:1172–1184, 1991.

[Heidemann et al., 2003] Heidemann, G., Rae, R., Bekel, H., Bax, I., Ritter, H. Integrating context-free and context-dependent attentional mechanisms for gestural object reference. In *Proc. International Conf. on Cognitive Vision Systems*, S. 22–33. Graz, Austria, 2003.

[Hellman, 1970] Hellman, M.E. The nearest neighbour classification rule with a reject option. *IEEE Trans. on Systems Science and Cybernetics*, 6:179–185, 1970.

[Henrichon und Fu, 1969] Henrichon, E.G., Fu, K.S. A nonparametric partitioning procedure for pattern classification. *IEEE Trans. on Computers*, 18:614–624, 1969.

[Hero et al., 2002] Hero, A.O., Ma, B., Michel, O.J.J., Gorman, J. Applications of entropic spanning graphs. *IEEE Signal Processing Magazine*, 19(5):85–95, 2002.

[Highleyman, 1962] Highleyman, W.H. The design and analysis of pattern recognition experiments. *Bell System Techn. Journal*, S. 723–744, 1962.

[Hillborn und Lainiotis, 1968] Hillborn, C.G., Lainiotis, D.G. Optimal unsupervised learning multicategory dependent hypotheses pattern recognition. *IEEE Trans. on Information Theory*, 14:468–470, 1968.

[Hinton, 1990] Hinton, G.E. Connectionist learning procedures. In J.G. Carbonell, Hg., *Machine Learning: Paradigms and Methods*, S. 185–234. MIT Press, Cambridge, MA, 1990.

[Ho, 1998] Ho, T.K. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:832–844, 1998.

[Ho und Basu, 2002] Ho, T.K., Basu, M. Complexity measures of supervised classification problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:289–300, 2002.

[Ho et al., 1994] Ho, T.K., Hull, J.J., Srihari, S.N. Decision combination in multiple classifier design. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:66–75, 1994.

[Holmström und Hämäläinen, 1993] Holmström, L., Hämäläinen, A. The self-organizing reduced kernel estimator. In *Proc. IEEE Int. Conf. Neural Networks*, S. Vol. 1, 417–421, 1993.

[Hopper, 1999] Hopper, F. *Fuzzy Cluster Analysis*. J. Wiley, Chichester, 1999.

[Hornegger et al., 1999] Hornegger, J., Paulus, D., Niemann, H. Probabilistic modeling in computer vision. In B. Jähne, H. Haußecker, P. Geißler, Hg., *Handbook of Computer Vision and Applications*, Bd. 2, S. 817–854. Academic Press, San Diego, 1999.

[Hornik, 1993] Hornik, K. Some new results on neural network approximation. *Neural Networks*, 6:1069–1072, 1993.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[Hoti und Holmström, 2004] Hoti, F., Holmström, L. A semiparametric density estimation approach to pattern classification. *Pattern Recognition*, 37:409–419, 2004.

[Hsu und Lin, 2002] Hsu, C.-W., Lin, C.-J. A comparison of methods for multiclass support vector machines. *IEEE Trans. on Neural Networks*, 13:415–425, 2002.

[Huang, 2003] Huang, G.-B. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Trans. on Neural Networks*, 14:274–281, 2003.

[Huang und Suen, 1995] Huang, T.S., Suen, C.Y. Combination of multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:90–94, 1995.

[Hussain, 1972] Hussain, A.B.S. On the correctness of some sequential classification schemes in pattern recognition. *IEEE Trans. on Computers*, 21:318–320, 1972.

[Hussain, 1974] Hussain, A.B.S. Compound sequential probability ratio test for the classification of statistically dependent patterns. *IEEE Trans. on Computers*, 23:398–410, 1974.

[Ichino, 1979] Ichino, M. A nonparametric multiclass pattern classifier. *IEEE Trans. on Systems, Man, and Cybernetics*, 9:345–352, 1979.

[Imai und Shimura, 1976] Imai, T., Shimura, M. Learning with probabilistic labeling. *Pattern Recognition*, 8:225–241, 1976.

[Ishibuchi et al., 1993] Ishibuchi, H., Fujioka, R., Tanaka, H. Neural networks that learn from fuzzy if-then rules. *IEEE Trans. on Fuzzy Systems*, 1:85–97, 1993.

[Itakura, 1975] Itakura, F. Minimum predicition residual principle applied to speech recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 23:67–72, 1975.

[Jain und Dubes, 1988] Jain, A., Dubes, R. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, USA, 1988.

[Jain et al., 2000] Jain, A.K., Duin, R.P.W., Mao, J. Statistical pattern recognition: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[Jain und Ranganath, 1982] Jain, A.K., Ranganath, S. Image restoration and edge extraction based on 2-d stochastic models. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 1520–1523. Paris, 1982.

[Jaynes, 1982] Jaynes, E.T. On the rationale of maximum entropy models. *Proc. IEEE*, 70:939–952, 1982.

[Jensen, 1996] Jensen, F.V. *An introduction to Bayesian networks*. UCL Press, London, 1996. 14GI/mat 5.2-487.

[Jeon und Landgrebe, 1994] Jeon, B., Landgrebe, D.A. Fast Parzen density estimation using clustering-based branch and bound. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:950–954, 1994.

[Johnson und Hebert, 1999] Johnson, A., Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:433–449, 1999.

[Juang et al., 1997] Juang, B.H., Chou, W., Lee, C.H. Minimum classification error rate methods for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 5:257–265, 1997.

[Juang und Rabiner, 1985] Juang, B.H., Rabiner, L.R. A probabilistic distance measure for hidden Markov models. *AT&T Technical J.*, 64(2):391–408, 1985.

[Kalinke und von Seelen, 1996] Kalinke, T., von Seelen, W. Entropie als Maß des lokalen Informationsgehaltes von Bildern zur Realisierung einer Aufmerksamkeitssteuerung. In B. Jähne, P. Geißler, H. Haußecker, F. Hering, Hg., *DAGM Symposium Mustererkennung*, S. 627–634. (Springer, Heidelberg), Heidelberg, Germany, 1996.

[Kan und Srinath, 2002] Kan, C., Srinath, M.D. Invariant character recognition with Zernicke and orthogonal Fourier-Mellin moments. *Pattern Recognition*, 35:143–154, 2002.

[Kanal und Chandrasekaran, 1971] Kanal, L., Chandrasekaran, S. On dimensionality and sample size in statistical pattern classification. *Pattern Recognition*, 3:225–234, 1971.

[Kantardzic, 2003] Kantardzic, M. *Data Mining: Concepts, Models, Methods, and Algorithms*. J. Wiley, New York, 2003.

[Kapur und Kesavan, 1992] Kapur, J.N., Kesavan, H.K. *Entropy Optimization Principles with Applications*. Academic Press, San Diego, USA, 1992.

[Kashyap, 1979] Kashyap, R.L. Syntactic decision rules for recognition of spoken words and phrases using a stochastic automaton. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:154–163, 1979.

[Katagiri et al., 1998] Katagiri, S., Juang, B.-H., Lee, C.-H. Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method. *Proc. IEEE*, 86:2345–2373, 1998.

[Katkovnik und Shmulevich, 2002] Katkovnik, V., Shmulevich, I. Kernel density estimation with adaptive varying window size. *Pattern Recognition Letters*, 23:1641–1648, 2002.

[Kay, 1993] Kay, S.M. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice

Hall, Englewood Cliffs, NJ, 1993.

[Kay, 1998] Kay, S.M. *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice Hall, Englewood Cliffs, NJ, 1998.

[Kazakos und Cotsidas, 1980] Kazakos, D., Cotsidas, T. A decision theory approach to the approximation of discrete probability densities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:61–67, 1980.

[Keehn, 1965] Keehn, D.G. A note on learning for Gaussian properties. *IEEE Trans. on Information Theory*, 11:126–132, 1965.

[Khanna, 1990] Khanna, T. *Foundations of Neural Networks*. Addison Wesley, Reading MA, 1990.

[Kim et al., 2003] Kim, H.-C., Kim, D., Bang, S.Y. An efficient model order selection for PCA mixture model. *Pattern Recognition Letters*, 24:1385–1393, 2003.

[Kim und Yang, 1995] Kim, I.Y., Yang, H.S. An integrated approach for scene understanding based on Markov random field model. *Pattern Recognition*, 28:1887–1897, 1995.

[Kim und Un, 1988] Kim, N.S., Un, C.K. Deleted strategy for MMI-based HMM training. *IEEE Trans. on Speech and Audio Processing*, 6:299–303, 1988.

[Kirby und Miranda, 1996] Kirby, N.J., Miranda, R. Circular nodes in neural networks. *Neural Computation*, 8:390–402, 1996.

[Kitano, 1990] Kitano, H. Empirical study of the speed of convergence of neural network training using genetic algorithms. In *Proc. 8. National Conference on Artificial Intelligence*, S. 789–795. AAAI Press/The MIT Press, Menlo Park, Cambridge, London, 1990.

[Kittler et al., 1998] Kittler, J., Hatef, M., Duin, R.P.W., Matas, J. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

[Kittler und Illingworth, 1985] Kittler, J., Illingworth, J. Relaxation labeling algorithms – a review. *Image and Vision Computing*, 3:206–216, 1985.

[Kittler und Roli, 2000] Kittler, J., Roli, F., Hg. *Multiple Classifier Systems*, Bd. 1857 von *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, 2000.

[Kittler und Roli, 2001] Kittler, J., Roli, F., Hg. *Multiple Classifier Systems*, Bd. 2096 von *Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, 2001.

[Kohonen, 1982] Kohonen, T. Clustering, taxonomy, and topological maps of patterns. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. 114–128. München, F.R. of Germany, 1982.

[Kohonen, 1989] Kohonen, T. *Self-Organization and Associative Memory*. Springer, Berlin Heidelberg, 3. Aufl., 1989.

[Kohonen, 1990a] Kohonen, T. Improved versions of learning vector quantization. In *Proc. Int. Joint Conference on Neural Networks*, S. I 545–550. San Diego, CA, 1990a.

[Kohonen, 1990b] Kohonen, T. The self-organizing map. *Proc. IEEE*, 78:1464–1480, 1990b.

[Koontz und Fukunaga, 1972] Koontz, W.L.G., Fukunaga, K. A nonparametric valley-seeking technique for cluster analysis. *IEEE Trans. on Computers*, 21:171–178, 1972.

[Koontz et al., 1976] Koontz, W.L.G., Narendra, P.M., Fukunaga, K. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Trans. on Computers*, 25:936–944, 1976.

[Kraaijveld, 1996] Kraaijveld, M.A. A Parzen classifier with an improved robustness against deviations between training and test data. *Pattern Recognition Letters*, 17:679–689, 1996.

[Kramer, 1991] Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks. *Journ. of the American Institute of Chemical Engineers*, 37(2):233–243, 1991.

[Kreyszig, 1967] Kreyszig, E. *Statistische Methoden und ihre Anwendungen*, Kap. 15. Vandenhoeck u. Ruprecht, Göttingen, 1967.

[Kruskal, 1956] Kruskal, J.B. On the shortest spanning subtree and the travelling salesman problem. *Proc. American Mathematical Society*, 7:48–50, 1956.

[Kruskal, 1964a] Kruskal, J.B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964a.

[Kruskal, 1964b] Kruskal, J.B. Nonmetric multidimensional scaling: A numerical method. *Psychome-*

trika, 29:115–129, 1964b.

[Kuhn et al., 1994] Kuhn, R., Lazarides, A., Normandin, Y., Boursseau, J., Nöth, E. Applications of decision tree methodolody in speech recognition and understanding. In H. Niemann, R. De Mori, G. Hanrieder, Hg., *CRIM/FORWISS Workshop on Progress and Prospects of Speech Research and Technology*, S. 220–232. infix Verlag, Sankt Augustin, München, Germany, 1994.

[Kulkarni und Kanal, 1976] Kulkarni, A.V., Kanal, L.N. An optimization approach to hierarchical classifier design. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. 459–466. Coronado, CA, 1976.

[Kuncheva, 2002] Kuncheva, L.I. A theoretical study of six classifier fusion strategies. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(2):281–286, 2002.

[Kung, 1993] Kung, S.Y. *Digital Neural Networks*. Prentice Hall, New York, 1993.

[Kwok und Tsang, 2004] Kwok, J.T., Tsang, I.W. The pre-image problem in kernel methods. *IEEE Trans. on Neural Networks*, 15:1517–1525, 2004.

[Kwong et al., 1998] Kwong, S., He, Q.H., Man, K.F., Tang, K.S. A maximum model distance approach for HMM-based speech recognition. *Pattern Recognition*, 31:219–229, 1998.

[Lachenbruch und Mickey, 1968] Lachenbruch, P.A., Mickey, M.R. Estimation of error rates in discriminant analysis. *Technometrics*, 10:715–725, 1968.

[Laferte et al., 2000] Laferte, J.-M., Perez, P., Heitz, F. Discrete Markov image modeling and inference on the quadtree. *IEEE Trans. on Image Processing*, 9:390–404, 2000.

[Lakany et al., 1997] Lakany, H.M., Schukat-Talamazzini, E.G., Niemann, H., Ghonaimy, M.A.R. Object recognition from 2D images using Kohonen self-organized feature maps. *Pattern Recognition and Image Analysis*, 7:301–308, 1997.

[Lam und Suen, 1997] Lam, L., Suen, C.Y. Application of majority voting to pattern recognition: An analysis of the behavior and performance. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 27:553–567, 1997.

[Lambert et al., 1999] Lambert, C., Harrington, S., Harvey, C., Glodjo, A. Efficient online nonparametric kernel density estimation. *Algorithmica*, 25:37–57, 1999.

[Lanckriet et al., 2004] Lanckriet, G.R.G., Christianini, N., Barlett, P., El Ghaoui, L., Jordan, M.I. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[Lau et al., 1993] Lau, R., Rosenfeld, R., Roukos, S. Trigger-based language models: A maximum entropy approach. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. II, 45–48. Minneapolis, USA, 1993.

[Laux, 1998] Laux, H. *Entscheidungstheorie*. Springer, Berlin, 1998.

[Lee und Lewicki, 2002] Lee, T.-W., Lewicki, M.S. Unsupervised image classification, segmentation, and enhancement using ICA mixture models. *IEEE Trans. on Image Processing*, 11:270–279, 2002.

[Lee und Mangasarian, 2001a] Lee, Y.-J., Mangasarian, O.L. RSVM: Reduced support vector machines. In *SIAM Int. Conf. on Data Mining*, 2001a.

[Lee und Mangasarian, 2001b] Lee, Y.-J., Mangasarian, O.L. SSVM: A smooth support vector machine for classification. *Computational Optimization and Applications*, 20(1):5–22, 2001b.

[Leonardis und Bischof, 1998] Leonardis, A., Bischof, H. An efficient MDL-based construction of RBF networks. *Neural Networks*, 11:963–973, 1998.

[Lepistö et al., 2003] Lepistö, L., Kunttu, I., Autio, J., Visa, A. Classification of non-homogenous textures by combining classifiers. In *Proc. IEEE Int. Conference on Image Processing (ICIP)*, S. Vol. 1, 1981–1984. Barcelona, Spain, 2003.

[Leshno et al., 1993] Leshno, M., Lin, V., Pinkus, A., Shochen, S. Multilayer feedforward networks with a polynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.

[Levinson et al., 1983] Levinson, S.E., Rabiner, L.R., Sondhi, M.M. An introduction to the application

of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Systems Technical Journal*, 62(4):1035–1074, 1983.

[Li, 1995] Li, S.Z. *Markov Random Field Modeling in Computer Vision*. Springer, Tokyo, 1995.

[LI, 2001] LI, S.Z. *Markov Random Field Modeling in Image Analysis*. Computer Science Workbench. Springer, Tokyo Berlin, 2001.

[Liao et al., 2003] Liao, Y., Fang, S.-C., Nuttle, H.L.W. Relaxed conditions for radial-basis function networks to be universal approximators. *Neural Networks*, 16:1019–1028, 2003.

[Lim und Lee, 1990] Lim, Y.W., Lee, S.U. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23(9):935–952, 1990.

[Lin, 2001] Lin, C.-J. On the convergence of the decomposition method for support vector machines. *IEEE Trans. on Neural Networks*, 12:1288–1298, 2001.

[Lin, 2002] Lin, C.-J. A formal analysis of stopping criteria of decomposition methods for support vector machines. *IEEE Trans. on Neural Networks*, 13:1045–1052, 2002.

[Lin und Lin, 2003] Lin, K.-M., Lin, C.-J. A study on reduced support vector machines. *IEEE Trans. on Neural Networks*, 14:1449–1459, 2003.

[Lin et al., 2003] Lin, X., Yacoub, S., Burns, J., Simske, S. Performance analysis of pattern classifier combination by plurality voting. *Pattern Recognition Letters*, 24:1959–1969, 2003.

[Linhart und Zucchini, 1986] Linhart, H., Zucchini, W. *Model Selection*. Wiley, New York, USA, 1986.

[Lippman, 1987] Lippman, R. An introduction to computing with neural nets. *IEEE Signal Processing Magazine*, S. 4–22, 1987.

[Liu et al., 2001] Liu, C., Zhu, S.C., Shum, H.Y. Learning inhomogeneous Gibbs model of face by minimax entropy. In *Proc. Int. Conference on Computer Vision (ICCV)*, 2001.

[Liu et al., 1995] Liu, C.-S., Lee, C.-H., Chou, W., Juang, B.-H., Rosenberg, A.E. A study of minimum error discriminative training for speaker recognition. *Journal of the Acoustical Society of America*, 97(1):637–648, 1995.

[Loftsgaarden und Quesenbury, 1965] Loftsgaarden, D.O., Quesenbury, G.P. A nonparametric estimate of a multivariable density function. *Annals of Mathematical Statistics*, 36:1049–1051, 1965.

[Lowe, 1987] Lowe, D.G. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[MacQueen, 1967] MacQueen, J. Some methods for classification and analysis of multivariate observations. In L.M.L. Cam, J. Neyman, Hg., *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, S. Vol. 1, 281–296, 1967.

[Makhoul et al., 1989] Makhoul, A., El-Jaroudi, A., Schwartz, R. Formation of disconnected decision regions with a single hidden layer. In *Proc. Int. Joint Conference on Neural Networks*, S. Vol. I, 455–460. Washington DC, USA, 1989.

[Malthouse, 1998] Malthouse, E.C. Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Trans. on Neural Networks*, 9:165–173, 1998.

[Mangasarian und Musicant, 2001] Mangasarian, O.L., Musicant, D.R. Lagrangian support vector machines. *J. of Machine Learning Research*, 1:161–177, 2001.

[Mao und Jain, 1995] Mao, J., Jain, A.K. Artificial neural networks for feature extraction and multivariate data projections. *IEEE Trans. on Neural Networks*, 6:296–317, 1995.

[Markou und Singh, 2003a] Markou, M., Singh, S. Novelty detection: a review – part 1: statistical approaches. *Signal Processing*, 83:2481–2497, 2003a.

[Markou und Singh, 2003b] Markou, M., Singh, S. Novelty detection: a review – part 2: neural network based approaches. *Signal Processing*, 83:2499–2521, 2003b.

[Martin et al., 1990] Martin, S.C., Ney, H., Hamacher, C. Maximum entropy language modeling and the smoothing problem. *IEEE Trans. on Speech and Audio Processing*, 8:626–632, 1990.

[Martinez, 2002] Martinez, A.M. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:748–763, 2002.

[Martinez und Vitria, 2000] Martinez, A.M., Vitria, J. Learning mixture models using a genetic version of the EM algorithm. *Pattern Recognition Letters*, 21:759–769, 2000.

[McCulloch und Pitts, 1943] McCulloch, W., Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Mathematical Biophysics*, 5:115–133, 1943.

[Meisel und Michalopoulos, 1973] Meisel, W.S., Michalopoulos, D.A. A partitioning algorithm with application in pattern classification and the optimization of decision trees. *IEEE Trans. on Computers*, 23:93–108, 1973.

[Meyer-Brötz und Schürmann, 1970] Meyer-Brötz, G., Schürmann, J. *Methoden der automatischen Zeichenerkennung*. R. Oldenbourg, München, 1970.

[Michie et al., 1994] Michie, D., Spiegelhalter, D.J., Taylor, C.C. *Machine Learning, Neural and Statistical Classification*. Artificial Intelligence. Ellis Horwood, 1994.

[Micó et al., 1996] Micó, M.L., Oncina, J., Carrasco, C. A fast branch and bound nearest neighbor classifier in metric spaces. *Pattern Recognition Letters*, 17:731–739, 1996.

[Middleton, 1960] Middleton, D. *An Introduction to Statistical Communication Theory*. McGraw Hill, New York, 1960.

[Mika et al., 1999] Mika, S., Schölkopf, B., Smola, A.J., Müller, K.-R., Scholz, M., Rätsch, G. Kernel PCA and de-noising in feature spaces. In M.S. Kearns, S.A. Solla, D.A. Cohn, Hg., *Advances in Neural Information Processing Systems*, S. 536–542. MIT Press, Cambridge, MA, USA, 1999.

[Minsky und Papert, 1969] Minsky, M., Papert, S. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, Mass., 1969.

[Mizoguchi und Shimura, 1976] Mizoguchi, R., Shimura, M. Nonparametric learning without a teacher based on mode estimation. *IEEE Trans. on Computers*, 25:1109–1117, 1976.

[Modestino und Zhang, 1992] Modestino, J.W., Zhang, J. A Markov random field model-based approach to image interpretation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:606–615, 1992.

[Moody und Darken, 1989] Moody, J., Darken, C. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[Moore, 1979] Moore, R.K. A dynamic programming algorithm for the distance between two finite areas. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:86–88, 1979.

[More und Wright, 1993] More, J.J., Wright, S.J. *Optimization Software Guide*. Frontiers in applied mathematics, 14. SIAM, Philadelphia, 1993.

[Moreno-Seco et al., 2003] Moreno-Seco, F., Micó, L., Oncina, J. A modification of the LAESA algorithm for approximated k-NN classification. *Pattern Recognition Letters*, 24:47–53, 2003.

[Morgan und Scofield, 1991] Morgan, D.P., Scofield, C.L. *Neural Networks and Speech Processing*. Kluwer Academic Publishers, Dordrecht, 1991.

[Morovic et al., 2002] Morovic, J., Shaw, J., Sun, P.-L. A fast, non-iterative and exact histogram matching algorithm. *Pattern Recognition Letters*, 23:127–135, 2002.

[Müller et al., 2001] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12:181–201, 2001.

[Murino und Vernazza, 2001] Murino, V., Vernazza, G. Guest Editorial: Artificial neural networks for image analysis and computer vision. *Image and Vision Computing*, 19(9-10):583–584, 2001.

[Muroga, 1971] Muroga, S. *Threshold Logic and its Applications*. J. Wiley, New York, 1971.

[Murthy, 1965] Murthy, V.K. Estimation of probability density. *Annals of Mathematical Statistics*, 36:1027–1031, 1965.

[Murua, 2002] Murua, A. Upper bounds for error rates of linear combinations of classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:591–602, 2002.

[Myers et al., 1980] Myers, C.S., Rabiner, L.R., Rosenberg, A.E. Perfomance tradeoffs in dynamic time warping algorithms for isolated word recogniton. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 28:623–635, 1980.

[Naphade und Huang, 2002] Naphade, M.R., Huang, T.S. Extracting semantics from audio-visual con-

tent: The final frontier in multimedia retrieval. *IEEE Trans. on Neural Networks*, 13:793–810, 2002.

[Navia-Vázquez et al., 2001] Navia-Vázquez, A., Pérez-Cruz, F., Artés-Rodríguez, A., Figueiras-Vidal, A.R. Weighted least squares training of support vector classifiers leading to compact and adaptive schemes. *IEEE Trans. on Neural Networks*, 12:1047–1059, 2001.

[Neuhoff, 1975] Neuhoff, D.L. The Viterbi algorithm as an aid in text recognition. *IEEE Trans. on Information Theory*, 21:222–226, 1975.

[Ney und Kuhn, 1980] Ney, H., Kuhn, M.H. Cluster analysis for telephone line speaker recognition. In M. Kunt, F. de Coulon, Hg., *Signal Processing, Theories and Applications*, S. 609–613. North Holland, Amsterdam, 1980.

[Neyman und Pearson, 1933] Neyman, J., Pearson, E.S. On the problem of the most efficient tests of statistical hypotheses. *Phil. Trans. Royal Soc. London*, 231:289–337, 1933.

[Niemann, 1969] Niemann, H. *Begründung und Anwendung einer Theorie zur quantitativen Beschreibung und Erkennung von Mustern*. Dissertation, Techn. Universität Hannover, Hannover, 1969.

[Niemann, 1970] Niemann, H. Mustererkennung mit orthonormalen Reihenentwicklungen. *Nachrichtentechn. Zeitschrift*, 23:308–313, 1970.

[Niemann, 1971] Niemann, H. An improved series expansion for pattern recognition. *Nachrichtentechn. Zeitschrift*, 24:473–477, 1971.

[Niemann, 1974] Niemann, H. *Methoden der Mustererkennung*. Akademische Verlagsgesellschaft, Frankfurt, 1974.

[Niemann, 1978] Niemann, H. Unüberwachtes Lernen. In E. Triendl, Hg., *Bildverarbeitung und Mustererkennung*, Bd. 17 von *Informatik Fachberichte*, S. 3–20. Springer, Berlin, Heidelberg, 1978.

[Niemann, 1979] Niemann, H. Digital image analysis. In P. Stucki, Hg., *Advances in Digital Image Processing*, S. 77–122. Plenum Press, New York, 1979.

[Niemann, 1980] Niemann, H. Linear and nonlinear mapping of patterns. *Pattern Recognition*, 12:83–87, 1980.

[Niemann, 1983] Niemann, H. *Klassifikation von Mustern*. Springer; (zweite erweiterte Auflage 2003 im Internet: http://www5.informatik.uni-erlangen.de/MEDIA/nm/klassifikation-von-mustern/m00links.html), Berlin, Heidelberg, 1983.

[Niemann und Goppert, 1988] Niemann, H., Goppert, R. An efficient branch-and-bound nearest neighbour classifier. *Pattern Recognition Letters*, 7:67–72, 1988.

[Niemann und Weiss, 1979] Niemann, H., Weiss, J. A fast-converging algorithm for nonlinear mapping of highdimensional data to a plane. *IEEE Trans. on Computers*, 28:142–147, 1979.

[Niemann und Wu, 1993] Niemann, H., Wu, J.K. Neural network adaptive image coding. *IEEE Trans. on Neural Networks*, 4:615–627, 1993.

[Nilsson, 1965] Nilsson, N.J. *Learning Machines*. McGraw Hill, New York, 1965.

[Noda et al., 2002] Noda, H., Shirazi, M.N., Kawaguchi, E. MRF-based texture segmentation using wavelet decomposed images. *Pattern Recognition*, 35:771–782, 2002.

[Normandin et al., 1994] Normandin, I., Cardin, R., De Mori, R. High-performance connected digit recognition using maximum mutual information estimation. *IEEE Trans. on Speech and Audio Processing*, 2:299–311, 1994.

[Normandin und Morgera, 1991] Normandin, S., Morgera, S.D. An improved mmie training for speaker-independent, small vocabulary, continuous speech recognition. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 537–540. Toronto, 1991.

[Normandin, 1996] Normandin, Y. Maximum mutual information estimation of hidden Markov models. In C.-H. Lee, F.K. Soong, K.K. Paliwal, Hg., *Automatic Speech and Speaker Recognition*, S. 57–81. Kluwer, Norwell MA, 1996.

[Och, 2002] Och, F.J. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. Dissertation, Rheinisch-Westfälische Technische Hochschule, Fakultät für Mathematik, Informatik und Naturwissenschaften, Aachen, Germany, 2002.

[Och und Ney, 2002] Och, F.J., Ney, H. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. 40th Annual Meeting of the Association for Computational Linguistics*. ACL, Philadelphia, USA, 2002.

[Olsen und Fukunaga, 1973] Olsen, D.R., Fukunaga, K. Representation of nonlinear data surfaces. *IEEE Trans. on Computers*, 22:915–922, 1973.

[Orchard, 1991] Orchard, M.T. A fast nearest-neighbor search algorithm. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 2297–2300. Toronto, Canada, 1991.

[Osborne et al., 2000] Osborne, M., Presnell, B., Turlach, B. On the LASSO and its dual. *J. Computational and Graphical Statistics*, 9:319–337, 2000.

[Othman und Aboulnasr, 2003] Othman, H., Aboulnasr, T. A separable low complexity 2D HMM with application to face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1229–1238, 2003.

[Pal und Bezdek, 1995] Pal, N., Bezdek, J.C. On cluster validity for the fuzzy c-means model. *IEEE Trans. on Fuzzy Systems*, 3:370–379, 1995.

[Pal et al., 2005] Pal, N.R., Pal, K., Keller, J.M., Bezdek, J.C. A probabilistic fuzzy c-means clustering algorithm. *IEEE Trans. on Fuzzy Systems*, 13:517–529, 2005.

[Panjwani und Healey, 1995] Panjwani, D.K., Healey, G. Markov random field texture models for unsupervised segmentation of textured color images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17:939–954, 1995.

[Papageorgiou, 1991] Papageorgiou, M. *Optimierung*. Oldenbourg, München, 1991.

[Park und Sandberg, 1991] Park, J., Sandberg, I.W. Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–257, 1991.

[Park und Sandberg, 1993] Park, J., Sandberg, I.W. Approximation and radial-basis-function networks. *Neural Computation*, 5:305–316, 1993.

[Parzen, 1962] Parzen, E. On estimation of a probability density and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[Patrick, 1972] Patrick, E.A. *Fundamentals of Pattern Recognition*. Prentice Hall, Englewood Cliffs N.J., 1972.

[Patrick und Costello, 1970] Patrick, E.A., Costello, J.P. On unsupervised estimation algorithms. *IEEE Trans. on Information Theory*, 16:556–569, 1970.

[Patrick et al., 1970] Patrick, E.A., Costello, J.P., Monds, F.C. Decision directed estimation of a two class decision boundary. *IEEE Trans. on Computers*, 19:197–205, 1970.

[Patrick und Hancock, 1966] Patrick, E.A., Hancock, J.P. Nonsupervised sequential classification and recognition of patterns. *IEEE Trans. on Information Theory*, 12:362–372, 1966.

[Pavlidis et al., 2001] Pavlidis, P., Furey, T.S., Liberto, M., Haussler, D., Grundy, W.N. Promoter region-based classification of genes. In R.B. Altman, A.K. Dunker, L. Hunker, K. Lauderdale, T.E. Klein, Hg., *Proc. Pacific Symposium on Biocomputing 6 (http://psb.stanford.edu/psb-online)*, S. 151–164. (World Scientific Press, ISBN 981-02-4515-7), Mauna Lani, Hawaii, 2001.

[Payne und Meisel, 1977] Payne, H.J., Meisel, W.S. An algorithm for constructing optimal binary decision trees. *IEEE Trans. on Computers*, 28:905–916, 1977.

[Pearl, 1988] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[Pelillo und Refice, 1994] Pelillo, M., Refice, M. Learning compatibility coefficients for relaxation labeling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:933–945, 1994.

[Peng et al., 2003] Peng, J., Heisterkamp, D.R., Dai, H.K. LDA/SVM driven nearest neighbor classification. *IEEE Trans. on Image Processing*, 14:940–942, 2003.

[Peng et al., 2004] Peng, J., Heisterkamp, D.R., Dai, H.K. Adaptive quasiconformal kernel nearest neighbor classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:656–661, 2004.

[Piché, 1994] Piché, S.W. Steepest descent algorithms for neural network controllers and filters. *IEEE*

Trans. on Neural Networks, 5:198–212, 1994.

[Plagianakos et al., 2002] Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N. Deterministic nonmonotone strategies for efficient training of multilayer perceptrons. *IEEE Trans. on Neural Networks*, 13:1268–1284, 2002.

[Platt, 1998] Platt, J.C. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, A. Smola, Hg., *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, Mass., 1998.

[Platt, 1999] Platt, J.C. Probabilistic outputs for support vector machines and comparisons to regularized likelihoods. In A. Smola, P. Bartlett, B. and Schölkopf, Hg., *Advances in Large Margin Classifiers*. The MIT Press, Cambridge, Mass., 1999.

[Poli et al., 1991] Poli, R., Cagnoni, S., Livi, R., Coppini, G., Valli, G. A neural network expert system for diagnosing and treating hypertension. *IEEE Computer*, 24(3):64–71, 1991.

[Pösl, 1999] Pösl, J. *Erscheinungsbasierte statistische Objekterkennung*. Berichte aus der Informatik. Shaker Verlag, Aachen, Germany, 1999.

[Postaire und Vasseur, 1981] Postaire, J.G., Vasseur, C.P.A. An approximate solution to normal mixture identification with application to unsupervised pattern classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3:163–179, 1981.

[Prim, 1957] Prim, R.C. Shortest connection networks and some generalizations. *Bell Systems Technical Journal*, 36:1389–1401, 1957.

[Principe et al., 2000] Principe, J., Xu, D., Fisher, J. Information theoretic learning. In S. Haykin, Hg., *Unsupervised Adaptive Filtering*. J. Wiley, New York, 2000.

[Pujari, 2001] Pujari, A.K. *Data Mining Techniques*. University Press, Nancy, 2001.

[Purnell und Botha, 2002] Purnell, D.W., Botha, E.C. Improved generalization of MCE parameter estimation with application to speech recognition. *IEEE Trans. on Speech and Audio Processing*, 10:232–239, 2002.

[Rabiner und Juang, 1993] Rabiner, L., Juang, B.-H. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, N.J., 1993.

[Rabiner, 1988] Rabiner, L.R. Mathematical foundations of hidden Markov models. In H. Niemann, M. Lang, G. Sagerer, Hg., *Recent Advances in Speech Understanding and Dialog Systems*, Bd. 46 von *NATO ASI Series F*, S. 183–205. Springer, Berlin, 1988.

[Rabiner et al., 1979] Rabiner, L.R., Levinson, S.E., Rosenberg, A.E., Wilpon, J.G. Speaker-independent recognition of isolated words using clustering techniques. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 27:336–349, 1979.

[Rabiner und Schmidt, 1980] Rabiner, L.R., Schmidt, G.E. Application of dynamic time warping to connected digit recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 28:377–388, 1980.

[Ramasubramanian und Paliwal, 1990] Ramasubramanian, V., Paliwal, K.K. An efficient approximation-elimination algorithm for fast nearest neighbour search based on a spherical distance coordinate formulation. In L. Torres-Urgell, M.A. Lagunas-Hernandez, Hg., *Signal Processing V: Theories and Applications (Proc. EUSIPCO)*, S. 1323–1326. (North Holland, Amsterdam), Barcelona, Spain, 1990.

[Ramasubramanian und Paliwal, 2000] Ramasubramanian, V., Paliwal, K.K. Fast nearest-neighbor search algorithms based on approximation-elimination search. *Pattern Recognition*, 33:1497–1510, 2000.

[Rao, 1973] Rao, C.R., Hg. *Linear Statistical Inference and its Applications*. J. Wiley, New York, 1973.

[Raphael, 2001] Raphael, C. Coarse-to-fine dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:1379–1390, 2001.

[Rätsch et al., 2002] Rätsch, G., Mika, S., Schölkopf, B., Müller, K.-R. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Trans. on Neural Networks*, 24:1184–1199, 2002.

[Rauber et al., 2002] Rauber, A., Merkl, D., Dittenbach, M. The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Trans. on Neural Networks*, 13:1331–1341, 2002.

[Raudys, 1991] Raudys, S. On the effectiveness of Parzen window classifier. *Informatica*, 2(3):434–454, 1991.

[Raudys und Jain, 1991] Raudys, S., Jain, A.K. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:252–264, 1991.

[Raudys und Pikelis, 1980] Raudys, S., Pikelis, V. On dimensionality, sample size, classification error and complexity of classification algorithm in pattern recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:242–252, 1980.

[Raviv, 1967] Raviv, J. Decision making in Markov chains applied to the problem of pattern recognition. *IEEE Trans. on Information Theory*, 13:536–551, 1967.

[Regel, 1982] Regel, P. A module for acoustic-phonetic transcription of fluently spoken German speech. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-30(3):440–450, 1982.

[Rehg et al., 2003] Rehg, J.M., Pavlovic, V., Huang, T.S., Freeman, W.T. Guest editor's introduction to the special section on graphical models in computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:785–786, 2003.

[Reinhold, 2003] Reinhold, M. *Robuste, probabilistische, erscheinungsbasierte Objekterkennung*. Dissertation, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, Germany, 2003.

[Reinhold, 2004] Reinhold, M. *Robuste, probabilistische, erscheinungsbasierte Objekterkennung*, Bd. 10 von *Studien zur Mustererkennung*. Logos, Berlin, 2004.

[Reinhold et al., 2001] Reinhold, M., Paulus, D., Niemann, H. Appearance-based statistical object recognition by heterogeneous background and occlusions. In B. Radig, S. Florczyk, Hg., *Pattern Recognition. Proc. 23rd DAGM Symposium*, S. 254–261. (Springer LNCS 2191, Berling Heidelberg, ISBN 3-540-42596-9), München, Germany, 2001.

[Reinhold et al., 2005] Reinhold, M.P., Grzegorzek, M., Denzler, J., Niemann, H. Appearance-based recognition of 3-D objects by cluttered background and occlusions. *Pattern Recognition*, 38:739–753, 2005.

[Renyi, 1960] Renyi, A. On measures of entropy and information. In *Proc. 4. Berkeley Symposium on Mathematics, Statistics, and Probability*, S. 547–561, 1960.

[Reynolds, 1995] Reynolds, D.A. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17:91–108, 1995.

[Reynolds und Rose, 1995] Reynolds, D.A., Rose, R.C. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Trans. on Speech and Audio Processing*, 3:72–83, 1995.

[Ridella et al., 1997] Ridella, S., Rovetta, S., Zunino, R. Circular backpropagation networks for classification. *IEEE Trans. on Neural Networks*, 8:84–97, 1997.

[Riseman und Hanson, 1974] Riseman, E.M., Hanson, A.R. A contextual postprocessing system for error correction using binary n-grams. *IEEE Trans. on Computers*, 23:480–493, 1974.

[Rissanen, 1978] Rissanen, J. Modeling by shortest data description. *Automation*, 14:465–471, 1978.

[Rissanen, 1989] Rissanen, J. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.

[Ritter und Gaggenmeier, 1999] Ritter, G., Gaggenmeier, K. Automatic classification of chromosomes by means of quadratically asymmetric statistical distributions. *Pattern Recognition*, 32:997–1008, 1999.

[Ritter et al., 1995] Ritter, G., Gallegos, M.T., Gaggenmeier, K. Automatic context-sensitive karyotyping of human chromosomes based on elliptically symmetric statistical distributions. *Pattern Recognition*, 28:823–831, 1995.

[Ritter et al., 1991] Ritter, H., Martinetz, T., Schulten, K. *Neuronale Netze*. Addison-Wesley, Bonn, 2.

Aufl., 1991.

[Roberts, 1965] Roberts, L.G. Machine perception of three-dimensional solids. In J.T. Tippelt, D.A. Berkowitz, L.C. Clapp, C.J. Koester, A.v.d. Burgh, Hg., *Optical and Electro-Optical Information Processing*, S. 159–197. MIT Press, Cambridge, 1965.

[Roberts et al., 1998] Roberts, S.J., Husmeier, D., Rezek, I., Penny, W. Bayesian approaches to Gaussian mixture modeling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, 1998.

[Rosenberg et al., 1998] Rosenberg, A.E., Siohan, O., Parthasarathy, S. Speaker verification using minimum verification error training. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 105–108. Seattle, Washington, 1998.

[Rosenblatt, 1961] Rosenblatt, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington D.C., 1961.

[Rosenfeld et al., 1976] Rosenfeld, A., Hummel, R.A., Zucker, S.W. Scene labeling by relaxation operations. *IEEE Trans. on Systems, Man, and Cybernetics*, 6:420–433, 1976.

[Rosenfeld, 1996] Rosenfeld, R. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech & Language*, 10:187–228, 1996.

[Roth, 2001] Roth, V. Probabilistic discriminative kernel classifier for multi-class problems. In B. Radig, S. Florczyk, Hg., *Pattern Recognition. Proc. 23rd DAGM Symposium*, S. 246–253. (Springer LNCS 2191, Berling Heidelberg, ISBN 3-540-42596-9), München, Germany, 2001.

[Roth, 2004] Roth, V. The generalized LASSO. *IEEE Trans. on Neural Networks*, 15:16–28, 2004.

[Roth et al., 2003] Roth, V., Laub, J., Kawanabe, M., Buhmann, J.M. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1540–1551, 2003.

[Rubner et al., 2000] Rubner, Y., Tomasi, C., Guibas, L.J. A metric for distributions with applications to image databases. *Int. Journal of Computer Vision*, 40(2):99–121, 2000.

[Rumelhart und McClelland, 1986] Rumelhart, D.E., McClelland, J.L. *Parallel Distributed Processing: Explorations into the Microstructure of Cognition*. MIT Press, Cambridge, MA, 1986.

[Rumelhart und McClelland, 1988] Rumelhart, D.E., McClelland, J.L. *Parallel Distributed Processing*, Bd. 1: Foundations. The MIT Press, Cambridge, Mass., eighth Aufl., 1988.

[Sakoe und Chiba, 1979] Sakoe, H., Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 23:43–49, 1979.

[Sammon, 1969] Sammon, J.W. A nonlinear mapping for data structure analysis. *IEEE Trans. on Computers*, 18:401–409, 1969.

[Sammon, 1970] Sammon, J.W. Interactive pattern analysis and classification. *IEEE Trans. on Computers*, 19:594–616, 1970.

[Sarimveis et al., 2002] Sarimveis, H., Alexandridis, A., Tsekouras, G., Bafas, G. A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Industrial and Engineering Chemistry Research*, 41:751–759, 2002.

[Schapire, 1990] Schapire, R.E. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[Schapire und Singer, 1999] Schapire, R.E., Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[Schiele und Crowley, 1996a] Schiele, B., Crowley, J.L. Object recognition using multidimensional receptive field histograms. In B.F. Buxton, R. Cipolla, Hg., *Proc. European Conference on Computer Vision (ECCV)*, S. Vol. 1, 610–619. (Springer LNCS 1064, Berlin Heidelberg), Cambridge, UK, 1996a.

[Schiele und Crowley, 1996b] Schiele, B., Crowley, J.L. Probabilistic object recognition using multidimensional receptive field histograms. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. Vol. B, 50–54. Wien, Austria, 1996b.

[Schiele und Crowley, 2000] Schiele, B., Crowley, J.L. Recognition without correspondence using multidimensional receptive field histograms. *Int. Journal of Computer Vision*, 36(1):31–52, 2000.

[Schölkopf, 1997] Schölkopf, B. *Support Vector Learning*. GMD-Bericht Nr. 287. Oldenbourg, München, 1997.

[Schölkopf et al., 2000] Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.I. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

[Schroeter und Bigün, 1995] Schroeter, P., Bigün, J. Hierarchical image segmentation by multi-dimensional clustering and orientation-adaptive boundary refinement. *Pattern Recognition*, 28:695–709, 1995.

[Schukat-Talamazzini, 1995] Schukat-Talamazzini, E.G. *Automatische Spracherkennung*. Vieweg, Wiesbaden, 1995.

[Schürmann, 1971] Schürmann, J. Über systematisch konstruierte nichtlineare Klassifikatoren für die Handblockschrift-Erkennung. *Elektron. Rechenanlagen*, 13:250–260, 1971.

[Schürmann, 1977] Schürmann, J. *Polynomklassifikatoren für die Zeichenerkennung*. R. Oldenbourg, München, 1977.

[Schürmann, 1978] Schürmann, J. A multifont word recognition system for postal address reading. *IEEE Trans. on Computers*, 27:721–732, 1978.

[Schürmann, 1982] Schürmann, J. Reading machines. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. 1031–1044. München, Germany, 1982.

[Schürmann und Becker, 1978] Schürmann, J., Becker, D. Spracherkennung mit Quadratmittel-Polynomklassifikatoren. *Elektron. Rechenanlagen*, 20:15–23, 65–71, 1978.

[Schürmann und Krause, 1974] Schürmann, J., Krause, P. Vergleich zweier quadratischer Klassifikatoren am gleichen Datenmaterial. *Elektron. Rechenanlagen*, 16:132–142, 1974.

[Schwarz et al., 1968] Schwarz, H., Rutishauser, H., Stiefel, E. *Numerik symmetrischer Matrizen*. B.G. Teubner, Stuttgart, 1968.

[Scott, 1992] Scott, D.W. *Multivariate Density Estimation*. J. Wiley, New York, 1992.

[Scudder, 1965] Scudder, H.J. Adaptive communication receivers. *IEEE Trans. on Information Theory*, 11:167–174, 1965.

[Sebestyen, 1962] Sebestyen, G. *Decision Making Processes in Pattern Recognition*. MacMillan, New York, 1962.

[Sebestyen und Edie, 1966] Sebestyen, G., Edie, J. An algorithm for nonparametric pattern recognition. *IEEE Trans. on Electronic Computers*, 15:908–915, 1966.

[Serratosa und Sanfeliu, 2006] Serratosa, F., Sanfeliu, A. Signatures versus histograms: Definitions, distances, and algorithms. *Pattern Recognition*, 39:921–934, 2006.

[Sethi und Chatterjee, 1977] Sethi, I.K., Chatterjee, B. Efficient decision tree design for discrete variable pattern recognition problems. *Pattern Recognition*, 9:197–206, 1977.

[Shapiro und Haralick, 1979] Shapiro, L.G., Haralick, R.M. Decomposition of two-dimensional shapes by graph-theoretic clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1:10–20, 1979.

[Shawe-Taylor und Cristianini, 2004] Shawe-Taylor, J., Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.

[Shepard, 1962] Shepard, R.N. The analysis of proximities: Multidimensional scaling with an unknown distance function I and II. *Psychometrika*, 27:125–140 and 219–246, 1962.

[Silverman, 1978] Silverman, B.W. Choosing the window width when estimating a density. *Biometrica*, 65:1–11, 1978.

[Silverman, 1985] Silverman, B.W. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, USA, 1985.

[Silverman, 1986] Silverman, B.W. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[Simons et al., 1997] Simons, M., Ney, H., Martin, S.C. Distant bigram language modeling using maximum entropy. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. II, 787–790. (IEEE Computer Society Press, Los Alamitos, CA), Munich, Germany, 1997.

[Siohan et al., 1998] Siohan, O., Rosenberg, A.E., Parthasarathy, S. Speaker identification using minimum classification error training. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 109–112. Seattle, Washington, 1998.

[Sirlantzkis et al., 2001] Sirlantzkis, K., Fairhurst, M.C., Hoque, M.S. Genetic algorithm for multiple classifier configuration: A case study in character recognition. In J. Kittler, F. Roli, Hg., *Proc. 2nd International Workshop on Multiple Classifier Systems*, S. 99–108. (Springer, LNCS 2096), 2001.

[Smith, 1994] Smith, S.J. Handwritten character classification using nearest neighbor in large databases. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:915–919, 1994.

[Spragins, 1966] Spragins, J. Learning without a teacher. *IEEE Trans. on Information Theory*, 12:223–230, 1966.

[Sproull, 1991] Sproull, R.F. Refinements to nearest-neighbour searching in k-dimensional trees. *Algorithmica*, 6:579–589, 1991.

[Srivastava et al., 2002] Srivastava, A., Liu, X., Grenander, U. Universal analytical forms for modeling image probabilities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1200–1214, 2002.

[Stein und Medioni, 1992] Stein, F., Medioni, G. Structural indexing: Efficient 3-D object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:125–145, 1992.

[Steinwart, 2003] Steinwart, I. On the optimal parameter choice for ν-support vector machines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1274–1284, 2003.

[Stephens, 2000] Stephens, M. Bayesian analysis of mixtures with an unknown number of components – an alternative to reversible jump methods. *The Annals of Statistics*, 28:40–74, 2000.

[Stepp und Michalski, 1986] Stepp, R.E., Michalski, R. Conceptual clustering of structured objects. *Artificial Intelligence*, 28:43–69, 1986.

[Stoffel, 1974] Stoffel, J.C. A classifier design technique for discrete variable pattern recognition problems. *IEEE Trans. on Computers*, 23:428–441, 1974.

[Stoica und Selén, 2004] Stoica, P., Selén, Y. Model-order selection. *IEEE Signal Processing Magazine*, 21(4):36–47, 2004.

[Storvik und Dahl, 2000] Storvik, G., Dahl, G. Lagrangian-based methods for finding MAP solutions to MRF models. *IEEE Trans. on Image Processing*, 9:469–479, 2000.

[Suen et al., 1990] Suen, C.Y., Nadal, C., Legault, R., Mai, T.A., Lam, L. Recognition of handwritten numerals based on the concept of multiple experts. In *Proc. 1st International Workshop on Frontiers in Handwriting Recognition*, S. 131–144, 1990.

[Sun et al., 2003] Sun, Y., Paik, J., Koschan, A., Page, D.L., Abidi, M.A. Point fingerprint: A new 3-D object representation scheme. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33:712–717, 2003.

[Suykens und Vandewalle, 1999] Suykens, J.A.K., Vandewalle, J. Least-squares support vector machine classifiers. *Neural Proc. Letters*, 9:293–300, 1999.

[Swain und Ballard, 1991] Swain, M.J., Ballard, D.H. Color indexing. *Int. Journal of Computer Vision*, 7:11–32, 1991.

[Swain und Hauska, 1977] Swain, P.H., Hauska, H. The decision tree classifier – design and potential. *IEEE Trans. on Geoscience Electronics*, 15:142–147, 1977.

[Takiyama, 1978] Takiyama, R. A general method for training the committee machine. *Pattern Recognition*, 10:255–259, 1978.

[Takiyama, 1981] Takiyama, R. A two-level committee machine: A representation and a learning procedure for general piecewise linear discriminant functions. *Pattern Recognition*, 13:269–274, 1981.

[Tamura und Tateishi, 1997] Tamura, S., Tateishi, M. Capabilities of a four-layered neural network: Four layers versus three. *IEEE Trans. on Neural Networks*, 8:251–255, 1997.

[Tarsitano, 2003] Tarsitano, A. A computational study of several relocation methods for k-means algorithms. *Pattern Recognition*, 36:2955–2966, 2003.

[Tax et al., 2000] Tax, D.M.J., van Breukelen, M., Duin, R.P.W., Kittler, J. Combining multiple classi-

fiers by averaging or by multiplying? *Pattern Recognition*, 33:1475–1485, 2000.

[Teicher, 1961] Teicher, H. Identifiability of mixtures. *Annals of Mathematical Statistics*, 32:244–248, 1961.

[Teicher, 1963] Teicher, H. Identifiability of finite mixtures. *Annals of Mathematical Statistics*, 34:1265–1269, 1963.

[Therrien, 1989] Therrien, C.W. *Decision, Estimation, and Classification*. J. Wiley, New York, 1989.

[Tibshirani, 1996] Tibshirani, R. Regression shrinkage and selection via the LASSO. *J. Royal Statistical Society*, B 58(1):267–288, 1996.

[Tipping, 2000] Tipping, M.E. The relevance vector machine. In S. Solla, T. Leen, K.-R. Müller, Hg., *Proc. Advances in Neural Information Processing Systems 12*, S. 652–658. MIT Press, Cambridge, MA, USA, 2000.

[Tipping, 2001] Tipping, M.E. Sparse Bayesian learning and the relevance vector machine. *J. of Machine Learning Research*, 1:211–244, 2001.

[Titsias und Likas, 2001] Titsias, M.K., Likas, A.C. Shared kernel models for class conditional density estimation. *IEEE Trans. on Neural Networks*, 12(5):987–997, 2001.

[Titsias und Likas, 2003] Titsias, M.K., Likas, A.C. Class conditional density estimation using mixtures with constrained component sharing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:924–928, 2003.

[Todd, 1962] Todd, J. *Survey of Numerical Analysis*. McGraw Hill, New York, 1962.

[Tomek, 1976] Tomek, I. Two modifications of cnn. *IEEE Trans. on Systems, Man, and Cybernetics*, 6:769–772, 1976.

[Toussaint, 1978] Toussaint, G. T. The use of context in pattern recognition. *Pattern Recognition*, 10:189–204, 1978.

[Toussaint, 1974] Toussaint, G.T. Bibliography on estimation of misclassification. *IEEE Trans. on Information Theory*, 20:472–479, 1974.

[Toussaint und Donaldson, 1970] Toussaint, G.T., Donaldson, R.W. Algorithms for recognizing contour-traced handprinted characters. *IEEE Trans. on Computers*, 19:541–546, 1970.

[Trazegnies et al., 2003] Trazegnies, C.de, Urdiales, C., Bandera, A., Sandoval, F. 3D object recognition based on curvature information of planar views. *Pattern Recognition*, 36:2571–2584, 2003.

[Tryon und Bailey, 1970] Tryon, R.C., Bailey, D.E. *Cluster Analysis*. McGraw Hill, New York, 1970.

[Tsypkin, 1973] Tsypkin, Y.Z. *Foundations of the Theory of Learning Systems*. Academic Press, New York, 1973.

[Ueda et al., 2000] Ueda, N., Nakano, R., Gharahmani, Y.Z., Hinton, G.E. SMEM algorithm for mixture models. *Neural Computation*, 12:2109–2128, 2000.

[Ulrich et al., 2003] Ulrich, M., Steger, C., Baumgartner, A. Real-time object recognition using a modified generalized Hough transform. *Pattern Recognition*, 36:2557–2570, 2003.

[Valiant, 1984] Valiant, L.G. A theory of the learnable. *Comm. Association for Computing Machinery*, 27:1134–1142, 1984.

[Van Campenhout, 1978] Van Campenhout, J.M. On the peaking of the Hughes mean recognition accuracy; the resolution of an apparent paradox. *IEEE Trans. on Systems, Man, and Cybernetics*, 8:390–395, 1978.

[Van Campenhout und Cover, 1981] Van Campenhout, J.M., Cover, T.M. Maximum entropy and conditional probability. *IEEE Trans. on Information Theory*, 27:483–489, 1981.

[Vanderheydt et al., 1980] Vanderheydt, L., Oosterlink, A., van Daele, J., Van Den Berghe, H. Design of a graph-representation and a fuzzy-classifier for human chromosomes. *Pattern Recognition*, 12:201–210, 1980.

[VanTrees, 1968] VanTrees, H.L. *Detection, Estimation, and Modulation Theory*. J. Wiley, New York, 1968.

[Vapnik, 1995] Vapnik, V.N. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[Vapnik, 1998] Vapnik, V.N. *Statistical Learning Theory*. J. Wiley, New York, 1998.

[Vapnik, 1999] Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. on Neural Networks*, 10:988–999, 1999.

[Veltkamp et al., 2000] Veltkamp, R.C., Burkhardt, H., Kriegel, H.-P. *State-of-the-Art in Content Based Image and Video Retrieval*. Kluwer Academic Publ., Boston, Dordrecht, London, 2000.

[Viterbi, 1967] Viterbi, A.J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Information Theory*, 13:260–269, 1967.

[Wald, 1939] Wald, A. Contributions to the theory of statistical estimation and testing of hypotheses. *Annals of Mathematical Statistics*, 10:299–326, 1939.

[Wald, 1950] Wald, A. *Statistical Decision Functions*. J. Wiley, New York, 1950.

[Wald, 1957] Wald, A. *Sequential Analysis*. J. Wiley, New York, 1957.

[Wallace und Boulton, 1968] Wallace, C.S., Boulton, D.M. An information measure for classification. *Computer Journal*, 11(2):195–209, 1968.

[Wand und Jones, 1995] Wand, M.P., Jones, M. *Kernel Smoothing*. Chapman and Hall, New York, 1995.

[Wang et al., 2006] Wang, J., Neskovic, P., Cooper, L.N. Neigborhood size selection in the k-nearest-neighbor rule using statistical confidence. *Pattern Recognition*, 39:417–423, 2006.

[Wang und Hirschberg, 1992] Wang, M.Q., Hirschberg, J. Automatic classification of intonational phrase boundaries. *Computer Speech & Language*, 6:175–196, 1992.

[Wasan, 1969] Wasan, M.T. *Stochastic Approximation*. Cambridge University Press, Cambridge, 1969.

[Webb, 2000] Webb, A.R. Gamma mixture models for target recognition. *Pattern Recognition*, 33(12):2045–2054, 2000.

[Wei und Hirzinger, 1994] Wei, G.Q., Hirzinger, G. Learning shape from shading by neural networks. In W.G. Kropatsch, H. Bischof, Hg., *Mustererkennung 1994; 16. DAGM Symposium und 18. Workshop der ÖAGM*, S. 135–144. PRODUserv, Berlin, 1994.

[Wei, 2002] Wei, J. Color object indexing and retrieval in digital libraries. *IEEE Trans. on Image Processing*, 11:912–922, 2002.

[Weidong und Zheng, 1986] Weidong, K., Zheng, H. Fast search algorithms for vector quantization. In *Proc. Int. Conference on Pattern Recognition (ICPR)*, S. 1007–1009. Paris, France, 1986.

[Whindham und Cutler, 1992] Whindham, M., Cutler, A. Information ratios for validating mixture analysis. *Journ. of the Am. Statistical Association*, 87:1188–1192, 1992.

[White, 1990] White, H. Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, 3:535–549, 1990.

[Wightman und Ostendorf, 1992] Wightman, C.W., Ostendorf, M. Automatic recognition of intonational features. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. Vol. 1, 221–224. San Francisco, USA, 1992.

[Williams et al., 2005] Williams, O., Blake, A., Cipolla, R. Sparse Bayesian learning for efficient visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27:1292–1304, 2005.

[Williams, 1995] Williams, P. Bayesian regularization and pruning using a Laplacian prior. *Neural Computation*, 7:117–143, 1995.

[Winkler, 1995] Winkler, G. *Image Analysis, Random Fields, and Dynamic Monte Carlo Methods*, Bd. 27 von *Applications of Mathematics*. Springer, Heidelberg, 1995.

[Wolfe, 1967] Wolfe, J.H. Normix, computational methods for estimating the parameters of multivariate normal mixtures of distributions. Res. Memo. SRM68-2, US Naval Personnel Research Activity, San Diego, CA, USA, 1967.

[Wolfe, 1970] Wolfe, J.H. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, 5:329–350, 1970.

[Wolfertstetter und Ruske, 1995] Wolfertstetter, F., Ruske, G. Structured Markov models for speech recognition. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, S. 544–547. Detroit, USA, 1995.

[Wolpert, 1992] Wolpert, D. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[Wu und Yang, 2006] Wu, K.-L., Yang, M.-S. Alternative learning vector quantization. *Pattern Reco-*

gnition, 39:351–362, 2006.

[Xu et al., 1992] Xu, L., Krzyzak, A., Suen, C.Y. Methods for combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. on Systems, Man, and Cybernetics*, 22:418–435, 1992.

[Yakowitz, 1970] Yakowitz, S.J. Unsupervised learning and the identification of finite mixtures. *IEEE Trans. on Information Theory*, 16:330–338, 1970.

[Yakowitz und Spragins, 1968] Yakowitz, S.J., Spragins, J. On the identifiability of finite mixtures. *Annals of Mathematical Statistics*, 39:209–214, 1968.

[Yamamoto, 1979] Yamamoto, H. A method of deriving compatibility coefficients for relaxation operators. *Computer Graphics and Image Processing*, 10:256–271, 1979.

[Yamany und Farag, 2002] Yamany, S.M., Farag, A.A. Surface signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:1105–1120, 2002.

[Yang und Wu, 2004] Yang, M.-S., Wu, K.-L. A similarity-based robust clustering method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:434–448, 2004.

[Yang und Liu, 2002a] Yang, X., Liu, J. Maximum entropy random fields for texture analysis. *Pattern Recognition Letters*, 23:93–101, 2002a.

[Yang und Liu, 2002b] Yang, X., Liu, J. Mixture density estimation with group membership functions. *Pattern Recognition Letters*, 23:501–512, 2002b.

[Yang und Zwolinski, 2001] Yang, Z.R., Zwolinski, M. Mutual information theory for adaptive mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:396–403, 2001.

[Yeung und Wang, 2002] Yeung, D.S., Wang, X.Z. Improving performance of similarity based clustering by feature weight learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:556–561, 2002.

[Yin, 2002] Yin, H. ViSOM - A novel method for multivariate data projection and structure visualization. *IEEE Trans. on Neural Networks*, 13:237–243, 2002.

[Z. et al., 2003] Z., Zhang., Chen, C., Sun, J., Chan, K.L. EM algorithm for Gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36:1973–1983, 2003.

[Zadeh, 1965] Zadeh, L.A. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[Zadeh, 1988] Zadeh, L.A. Fuzzy logic. *IEEE Computer*, 21(4):83–93, 1988.

[Zadeh et al., 1975] Zadeh, L.A., Fu, K.S., Tanaka, K., Shimura, M., Hg. *Fuzzy Sets and Their Application to Cognitive and Decision Processes*. Academic Press, New York, 1975.

[Zahn, 1971] Zahn, C.T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. on Computers*, 20:68–86, 1971.

[Zell, 1994] Zell, A. *Simulation Neuronaler Netze*. Addison-Wesley (Deutschland) GmbH, Bonn, 1994.

[Zhang et al., 2004] Zhang, B., Zhang, C., Yi, X. Competitive EM algorithm for finite mixture models. *Pattern Recognition*, 37:131–144, 2004.

[Zhang et al., 2006] Zhang, D., Chen, S., Zhou, Z.-H. Learning the kernel parameters of kernel minimum distance classifier. *Pattern Recognition*, 39:133–135, 2006.

[Zhang und Benveniste, 1992] Zhang, Q., Benveniste, A. Wavelet networks. *IEEE Trans. on Neural Networks*, 3:889–898, 1992.

[Zhu und Hastie, 2005] Zhu, J., Hastie, T. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14(1):185–205, 2005.

[Zhu, 2003] Zhu, S.C. Statistical modeling and conceptualization of visual patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:691–712, 2003.

[Zhu et al., 2000] Zhu, S.C., Liu, X.W., Wu, Y.N. Exploring texture ensembles by efficient Markov chain Monte Carlo – toward a "trichromacy" theory of texture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:554–569, 2000.

[Zhu et al., 1997] Zhu, S.C., Wu, Y.N., Mumford, D.B. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9:1627–1660, 1997.

[Zhu et al., 1998] Zhu, S.C., Wu, Y.N., Mumford, D.B. Filters, random fields, and maximum entropy (FRAME): Towards a unified theory of texture modeling. *Int. Journal of Computer Vision*, 27(2):1–20, 1998.

[Zhuravlev, 1976] Zhuravlev, Y.I. Non-parametric problems of pattern recognition. *Kibernetika*, (6):93–123, 1976.

[Zhuravlev, 1977a] Zhuravlev, Y.I. Correct algebras over a set of inaccurate (heuristic) algorithms i. *Kibernetika*, (4):14–21, 1977a.

[Zhuravlev, 1977b] Zhuravlev, Y.I. Correct algebras over a set of inaccurate (heuristic) algorithms ii. *Kibernetika*, (6):21–27, 1977b.

[Zhuravlev, 1978a] Zhuravlev, Y.I. Correct algebras over a set of inaccurate (heuristic) algorithms iii. *Kibernetika*, (2):35–43, 1978a.

[Zhuravlev, 1978b] Zhuravlev, Y.I. On the algebraic approach to the solution of problems of recognition and classification. In *Problemy Kibernetiki (Problems of Cybernetics), Bd. 33*, S. 5–68. Nauka, Moscow, 1978b.

[Zhuravlev und Gurevich, 1991] Zhuravlev, Y.I., Gurevich, I.B. Pattern recognition and image recognition. *Pattern Recognition and Image Analysis*, 1:149–181, 1991.

[Zucker et al., 1977] Zucker, S.W., Hummel, R.A., Rosenfeld, A. An application of relaxation labeling to line and curve enhancement. *IEEE Trans. on Computers*, 26:394–403, 1977.

Index