

Context-Dependent Output Densities for Hidden Markov Models in Speech Recognition

Georg Stemmer, Viktor Zeissler, Christian Hacker, Elmar Nöth, Heinrich Niemann

Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, 91058 Erlangen, Germany
stemmer@informatik.uni-erlangen.de

Abstract

In this paper we propose an efficient method to utilize context in the output densities of HMMs. State scores of a phone recognizer are integrated into the HMMs of a word recognizer which makes their output densities context-dependent. A significant reduction of the word error rate has been achieved when the approach is evaluated on a set of spontaneous speech utterances. As we can expect that context is more important for some phone models than for others, we further extend the approach by state-dependent weighting factors which are used to control the influence of the different information sources. A small additional improvement has been achieved.

1. Introduction

A well-known weakness in HMMs is that the feature vectors are dependent only on the states which generated them, not on the neighboring feature vectors. Context is only represented by the dynamic features, e.g. delta coefficients of the Mel-frequency cepstral coefficients. However, most types of dynamic features are only limited to a few subsequent feature vectors and do not represent long-term variations. In [1] we have introduced a method to incorporate context into HMMs by simply taking into account a phone recognizer, which runs in parallel to the word recognizer. The state scores of the phone recognizer are computed with the beam search algorithm. They depend on all feature vectors that have been observed so far; the fact that the HMMs of the phone recognizer are based on the Markov assumption is not relevant. This makes the state scores of the phone recognizer a valuable additional information source for each state of the word recognizer.

We can assume that context is more useful for the recognition of certain sounds than others. For instance, we expect that the recognition of vowels like /E/ does not need much context while for short sounds, e.g. plosives like /p/ context could be very important (phone transcriptions in this paper are in SAMPA, see <http://www.phon.ucl.ac.uk/home/sampa.htm>). It has also been shown in the literature [2] that the importance of dynamic features for the recognition accuracy differs between the phones. After a review of the mathematical formalism we will extend the approach from [1] by state-dependent weighting factors which are used to control the influence of the different information sources. For each state of a phone model the influence of the context on the output density is determined individually.

The most successful ways to enhance the use of context in HMMs that can be found in literature are based on improvements of the extraction of temporal features [3], but this is be-

yond the scope of this paper. A number of studies to overcome the so-called conditional independence assumption of HMMs based on an improvement of the *model* are described in [4]. The concept of segment models is also related to this topic, please refer to [5] for an overview. Most of the approaches perform direct modeling of segments of speech frames, others assume that the output distribution of the HMM does not only depend on the current state but also on one or several previous frames [4]. A major disadvantage of most of these methods is that the parameter space increases dramatically, even if only one neighboring feature vector is considered.

The number of free parameters can be reduced by representing the context with a discrete random variable ([6], p. 409). This is similar to the approach described in this paper, as the context is also represented by a single discrete random variable. However, the context is not limited to a few feature vectors and the computation scheme for the output distribution has much less free parameters. Another major advantage of the approach introduced below is that the algorithms for training and decoding are not changed, so there is no increase in the complexity of the computation.

2. Mathematical Formalism

2.1. Output Density

In a standard (semi-)continuous HMM the density function $b_i(\mathbf{x}_t)$ for the output of a feature vector \mathbf{x}_t by the state i at time t is computed by a sum over all codebook classes $m \in M$:

$$b_i(\mathbf{x}_t) = \sum_m c_{i,m} \cdot p(\mathbf{x}_t|m, i) \approx \sum_m c_{i,m} \cdot p(\mathbf{x}_t|m) \quad (1)$$

The probability for a certain codebook class m , given a state i is represented by $c_{i,m}$. The second part in Eq. 1 corresponds to the transition from continuous to semi-continuous HMMs. A Gaussian pdf $\mathcal{N}(\mathbf{x}_t|\mu_m, \Sigma_m)$ is typically used to represent $p(\mathbf{x}_t|m)$.

In the rest of this paper we will consider probability density functions which make it possible to integrate a large context \mathbf{x}_1^{t-1} into the HMM output density. \mathbf{x}_1^{t-1} stands for the context $\mathbf{x}_1, \dots, \mathbf{x}_{t-1}$ of feature vectors which have been observed so far. If we try to integrate the context \mathbf{x}_1^{t-1} directly into b_i this results in a large amount of additional computational effort.

Therefore we introduce a new hidden random variable l , which we call the class label. Each of the class labels $l \in L$ may correspond to a phone symbol, for instance. From now on each state i does not only choose between the codebook classes $m \in M$, but at the same time also takes an independent decision for the class label l . The class label l itself is a discrete

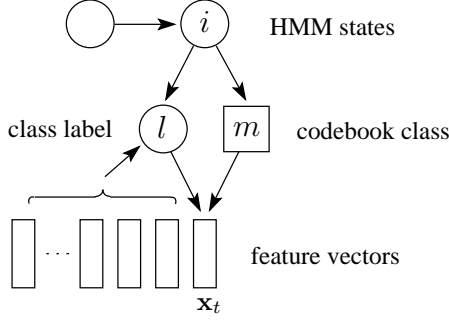


Figure 1: Output of the feature vector \mathbf{x}_t by the HMM state i . The arrows symbolize statistical dependencies between random variables, not state transitions.

representation of the complete history of feature vectors \mathbf{x}_1^{t-1} . The integration of l into the output density makes b_i dependent on the history \mathbf{x}_1^{t-1} . Unlike the approaches which have been mentioned in the literature review, we do not entirely abandon the conditional independence assumption of HMMs: the new model still assumes that \mathbf{x}_t is independent from the history \mathbf{x}_1^{t-1} when l and m are known. The process of feature vector generation according to the new model is illustrated in Fig. 1. The probability term $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ has to be expanded as follows:

$$b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1}) = \sum_{l,m} p(\mathbf{x}_t|l, m, i) \cdot P(l, m|i, \mathbf{x}_1^{t-1}) \quad (2)$$

As \mathbf{x}_1^{t-1} is the same for all states i at time t , there is no increase in the computational complexity of the algorithms for training and decoding.

2.2. Simplifying Assumptions

The representation of $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ requires the estimation of too many parameters if we do not make additional simplifications. Firstly we can use the following approximation since the decisions for m and l are independent and m does not depend on \mathbf{x}_1^{t-1} :

$$P(l, m|i, \mathbf{x}_1^{t-1}) = c_{i,m} \cdot P(l|i, \mathbf{x}_1^{t-1}) \quad (3)$$

Secondly we can split $P(l|i, \mathbf{x}_1^{t-1})$ into two parts under the assumption that i is independent from \mathbf{x}_1^{t-1} :

$$P(l|i, \mathbf{x}_1^{t-1}) \propto P(l|i) \cdot P(l|\mathbf{x}_1^{t-1}) \quad (4)$$

$P(l|i)$ is estimated during the Baum-Welch training, while the computation of $P(l|\mathbf{x}_1^{t-1})$ is different for each type of class labels that are employed. Finally, we can compute the output density values of the models separately as m does not depend on l :

$$p(\mathbf{x}_t|l, m, i) \propto p(\mathbf{x}_t|m) \cdot p(\mathbf{x}_t|l) \quad (5)$$

To summarize, $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ is computed by

$$b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1}) \approx \left[\sum_m c_{i,m} \cdot p(\mathbf{x}_t|m) \right]^{w_i} \cdot \left[\sum_l P(l|i) \cdot P(l|\mathbf{x}_1^{t-1}) \cdot p(\mathbf{x}_t|l) \right]^{1-w_i} \quad (6)$$

The weighting factor w_i is introduced to control the influence of the different knowledge sources on $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$. In [1] a

global weighting factor w for all states was employed. Here we extend the approach by using state-dependent weighting factors w_i which are chosen for each HMM state individually. We will optimize the w_i on the development test set.

3. Weighting Factor Optimization

When all states i share the same global weighting factor w the optimization can be done by a simple grid search which evaluates the error rate of the recognizer for several values of w on the development test set. For the state-dependent weighting factors w_i we use a similar optimization method as the one which has been described in [2].

The w_i are optimized on a data set which is independent of the training set of the recognizer. For each utterance a Viterbi search is performed to determine the optimal state sequence for a given transcription. The optimization procedure tries to adapt the weights w_i in order to maximize the Viterbi score, i.e. the weights should increase the output probability $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ for all states i which are in the optimal path. In contrast to [2] we do not apply any methods to discriminate between HMM states, i.e. the weights w_j from states j which are not in the forced alignment path are not altered. Gradient ascent is applied to maximize the output probability $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$; the update rule for the weight w_i of state i is

$$\hat{w}_i = w_i + \rho \cdot \frac{d b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})}{d w_i} \quad (7)$$

The second term stands for the derivative of b_i with respect to w_i which is weighted by the step size ρ . As $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ is of the form

$$b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1}) = a_1^{w_i} \cdot a_2^{1-w_i} \quad (8)$$

we can compute the derivative of b_i with respect to w_i from

$$\frac{d b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})}{d w_i} = (a_1^{w_i} \cdot a_2^{1-w_i}) \cdot (\log a_1 - \log a_2) \quad (9)$$

The first factor in this product is always ≥ 0 therefore the weight w_i is increased for $a_1 > a_2$ and it is decreased for $a_2 > a_1$. In [2] the derivative of the log-likelihood is used which is $\log a_1 - \log a_2$. Both variants have always the same sign of the gradient, so this should make no great difference in practice.

As the data set which is used for the weighting factor optimization is quite small the states of phone models which have the same right biphone model get tied. For instance, the first state of n/a:/x shares the weighting factor with the first state of /a:/x.

4. Integration of the Phone Recognizer

4.1. Combination of Information Sources

The improvements which may be achieved with our approach depend to a large part on the specific choice of the class labels l and the corresponding density functions $p(\mathbf{x}_t|l) \cdot P(l|\mathbf{x}_1^{t-1})$. As we decided to use a phone recognizer as information source, the labels l represent states of phone models. The density value can be computed from the probability that the current state s_t of a phone HMM is equal to l :

$$p(\mathbf{x}_t|l) \cdot P(l|\mathbf{x}_1^{t-1}) := p(\mathbf{x}_t|l) \cdot P(s_t = l|\mathbf{x}_1^{t-1}) \quad (10)$$

where $P(s_t = l|\mathbf{x}_1^{t-1})$ is calculated from the forward score:

$$P(s_t = l|\mathbf{x}_1^{t-1}) = \frac{P(s_t = l, \mathbf{x}_1^{t-1})}{\sum_j P(s_{t-1} = j, \mathbf{x}_1^{t-1})} \quad (11)$$

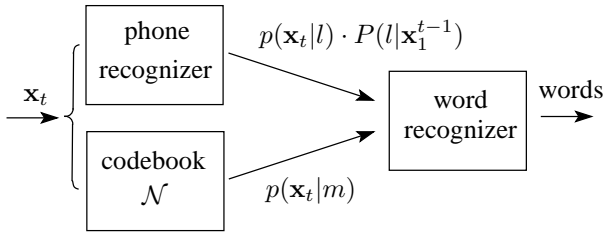


Figure 2: A phone recognizer as an information source for the word recognizer.

The latter is approximated by the Viterbi score of the state which is computed during beam search decoding in the phone recognizer. $p(\mathbf{x}_t|l)$ is the output density value of state l and is modeled as a mixture of Gaussian pdfs. The system architecture is illustrated in Fig. 2.

4.2. State Clustering

As the phone recognizer has about 300 different states (including the models for pauses, filled pauses and nonverbal sounds), we reduce the number of parameters by clustering similar states into groups. The use of state clusters for the class labels in contrast to individual states also increases robustness w.r.t. phone recognition errors.

A symmetric distance $D'(i, j)$ between two states i, j for semi-continuous HMMs can be computed from the Kullback-Leibler distance $D(i|j)$ of their output densities:

$$D(i|j) = \sum_m c_{i,m} \cdot \ln \frac{c_{i,m}}{c_{j,m}} \quad (12)$$

$$D'(i, j) = \frac{1}{2}D(i|j) + \frac{1}{2}D(j|i) \quad (13)$$

We apply the clustering algorithm from [7], p. 143: The size of a cluster C is defined as the maximum distance between any two states in C :

$$size(C) = \max_{i,j \in C} D'(i, j) \quad (14)$$

Initially each cluster contains exactly one state. The pair of clusters which when combined would form the smallest resultant cluster are merged. We repeat this step until the desired total number of clusters is reached.

In all experiments which are described below, the class label l stands not for a single HMM state but for a state cluster C_l . The probability of a specific label l is computed by averaging the scores of all states s_t which are in the same cluster C_l .

5. Data

Acoustic models are trained on a part of the EVAR data set. It consists of 7438 utterances, which have been recorded by phone with our conversational train timetable information system. A detailed description of this system can be found in [8]. Nearly all utterances are in German language. The total amount of data is ≈ 8 hours. 4999 utterances have randomly been selected for training, the development test set contains 441 utterances. The rest of 1998 utterances is available for testing. The speakers of the training and the test sets are disjoint.

clusters	5	10	15	20	25	30
WER[%]	25.3	25.0	24.7	24.6	24.7	25.1

Table 1: Comparison of the error rates when the total number of clusters is varied. w is set to 0.5 during training; for decoding w has been optimized on the development test set.

6. Baseline System

The system which has been used for the experiments is a speaker independent continuous speech recognizer. It is based on semi-continuous HMMs, the output densities of the HMMs are full-covariance Gaussians. Please refer to [8] for a detailed description of the speech recognizer. If the baseline system is only trained on the training data set described in the next section and no other data is used for training or initialization of the acoustic models, it achieves a word error rate of 26.0% on the test data.

7. Experimental Results

7.1. Training

The training of the whole system is done in three steps: Firstly the phone recognizer which generates the class labels l is trained. As we do not have a phone-level annotation of the training data, we simply replace each word in the transcription by its canonic phone representation. The phone recognizer achieves a phone error rate of 43.9% on the test data. Secondly we run the phone recognizer on the training data in order to compute the labels and the corresponding density values which are then used for the training of the word recognizer with the Baum-Welch algorithm. The weighting factor w is set to 0.5 for all states during the Baum-Welch training of the word recognizer. In the following, we will call the global weighting factor which is the same for all states w , while w_i stands for the state-dependent weighting factors. The optimal choice for the value of the global weighting factor w during decoding is determined on the development test set; w is varied in the range of 0.55–0.95 in steps of 0.05. We got slight improvements, when we additionally used the optimized weighting factor to re-train the word recognizer from scratch. Finally we estimate the state-dependent weighting factors w_i using the optimization procedure described in Sec. 3 on the development test set. During this phase no other parameters of the recognition system are changed. The step size ρ is set to 0.1 and similar to [2] only two iterations of the gradient ascent are performed.

7.2. Results and Discussion

In a first experiment the total number of clusters is varied. Two separate clusters are manually assigned to all states of HMMs for pauses and nonverbal sounds, the rest of the states is clustered with the algorithm described above. As shown in Tab. 1, the system performance is significantly better than the baseline for 15–25 state clusters. The optimal global weight factor w is for all numbers of clusters in the range between 0.8 and 0.9. We use 20 clusters (plus two for the pauses and nonverbal sounds) for all following experiments. We compare the word error rates (WER) for the experiments with context-dependent output densities to the baseline in Tab. 2. The WER for the phone recognizer with the global weighting factor w is slightly better than in Tab. 1, because we used the optimized global weight ($w = 0.8$ for all states i) for the training of the word recog-

experiment	WER [%]	relative improvement [%]
baseline	26.0	-
global w	24.4	6.2
state-dependent w_i	24.1	7.3

Table 2: Comparison of the experiments with global and state-dependent weighting factors for the context-dependent output densities and relative improvements over the baseline. The number of clusters is set to 20, all recognizers are trained with global weights w which have been optimized on the development test set.

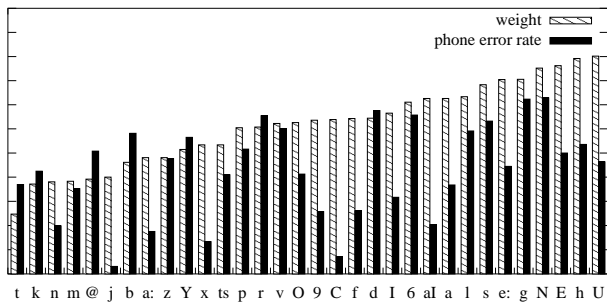


Figure 3: Phone error rates of the phone recognizer and weighting factors w_i for the most frequent phones in our dataset. For a better visualization, the scaling of the vertical axis is different for the phone error rates and the weighting factors. The phones are sorted with respect to their weighting factor, i.e. models of phones on the left make more use of the context information than the models of the phones which are more on the right.

nizer. The optimal global weight on the development test set is then $w = 0.85$. The state-dependent weighting factors give a small improvement over the global weighting factor. As the state-dependent optimized weighting factors are in a range between 0.77 and 0.90 the changes are probably just too small to have significant influence on the recognition rate. The best result corresponds to an improvement of 7% relative (1.9 percent points) over the baseline. The fact that the accuracy of a phone recognizer is typically quite low does not seem to hurt too much. It is possible that we have prevented this by using scores for state clusters and not the final output of the phone recognizer. The scores contain an interpolated representation of all possible hypotheses and not only the single best phone sequence.

The resulting state-dependent weighting factors could help us to get more insight into how the states of the word recognizer actually utilize the context information. Fig. 3 shows the weighting factors w_i for the most frequent phones in our dataset, averaged over all contexts and HMM states. For each phone also the corresponding phone error rate of the phone recognizer is given. As can be seen easily, the correlation between phone error rate and weighting factor is quite low. For each individual phone the optimal weighting factor seems to be a result of a combination of the phone error rate and the phone's properties. For instance, the plosives /t/, /k/, /b/, /p/ have all lower weighting factors than the average, i.e. models of plosives seem to make use of the context. However, the plosives /d/ and /g/ have larger weights as their corresponding phone error rate is very high. Outliers in Fig. 3 may be due to limitations of the task where some phones occur only in very few different con-

texts, the general tendency however makes sense. If the models for nonverbal sounds (i.e. background noise, clicks and others) would be inserted into Fig. 3 they would be placed between /j/ and /b/. Obviously context provides useful information to distinguish between verbal and nonverbal sounds.

8. Conclusion and Future Work

For most speech recognition systems, the only way to incorporate temporal context in the output distributions of the HMMs is to use dynamic features. We have introduced a new method to integrate context into the recognition process of a word recognizer: we simply use the state scores of a phone recognizer which runs in parallel to the word recognizer. Our experiments show that the proposed approach reduces the word error rate significantly.

Future work includes further investigations into the application of different types of sub-word units for the acoustic pre-processor. However, our approach is not limited to the use of states or state clusters for the class labels. Therefore we plan to evaluate if the performance of the word recognizer can be improved by using (context dependent) classifiers for other types of labels, e.g. for noisy or voiced/unvoiced speech.

9. Acknowledgments

A part of this work was funded by the German Federal Ministry of Education, Science, Research and Technology (*BMBF*) in the framework of the SmartKom project under Grant 01 IL 905 K7. The responsibility for the content lies with the authors.

10. References

- [1] G. Stemmer, V. Zeissler, C. Hacker, E. Nöth, and H. Niemann, "A phone recognizer helps to recognize words better," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003.
- [2] I. Rogina and A. Waibel, "Learning state-dependent stream weights for multi-codebook HMM speech recognition systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1994, pp. 217–220.
- [3] N. Morgan, "Temporal Signal Processing for ASR," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU'99)*, 1999.
- [4] J. Ming and F. J. Smith, "Modelling of the Interframe Dependence in an HMM Using Conditional Gaussian Mixtures," *Computer Speech and Language*, vol. 10, no. 4, pp. 229–247, 1996.
- [5] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 5, pp. 360–378, 1996.
- [6] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing*. Upper Saddle River, New Jersey: Prentice Hall, 2001.
- [7] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.0)*. Microsoft Corporation, 2000.
- [8] F. Gallwitz, *Integrated Stochastic Models for Spontaneous Speech Recognition*, ser. Studien zur Mustererkennung. Berlin: Logos Verlag, 2002, vol. 6.