Optimal projections between Gaussian Mixture Feature Spaces for Multilingual Speech Recognition

M. Raab[1,3], O. Schreiner[1,2], T. Herbig[1,2], R. Gruhn [1,2], E. Nöth[3]

[1] *Harman Becker Automotive Systems, Speech Dialog Systems, Ulm, Germany*

[2] *University of Ulm, Institute of Information Technology, Ulm, Germany*

[3] *University of Erlangen-Nuremberg, Chair of Pattern Recognition, Erlangen, Germany*

Introduction

Multilingual speech recognition is increasingly gaining attention for in-car speech controlled applications. An example is a media player that allows selection of music by voice command, requiring speech recognition for multiple languages in order to cover the languages of artist names and music titles in a given music database.

There are two traditional approaches how a system can support the recognition of multiple languages at the acoustic model level. The first is to run a set of monolingual recognizers in parallel; the second one is to train a multilingual recognizer for the required set of languages. The first approach has the disadvantage that there is no parameter sharing between the different recognizers, thus needing large amounts of processing power and memory. The second approach has the disadvantage that one multilingual recognizer has to be trained for every combination of languages.

In our paper we present a scheme to create a multilingual recognizer out of monolingual trained recognizers. A formula is given for an optimal projection of emission probabilities between Gaussians Mixture feature spaces. With this formula, we can project each HMM state of all languages to one set of Gaussians without retraining the acoustic model.

Multilingual Speech Recognition

Compared to monolingual speech recognition, multilingual speech recognition introduces some additional issues. A basic design question of multilingual systems is whether the system should recognize all languages as well as possible, or if the focus should be more on one main language, in which the performance is more important than in other languages. If the computational resources are limited, this question has to be answered in order to allocate different numbers of parameters for different languages. This limit is also the reason why it is not possible to just run monolingual recognizers for all languages in parallel. Work in the literature has mostly dealt with the first task, to provide multilingual speech recognition as well as possible for all languages. With this target, previous research has achieved parameter reductions by combining Hidden Markov Models (HMM) from different languages to one model, if they share the same International Phonetic Alphabet (IPA, [6]) symbol [12, 11]. There are also papers that compare the combination of models based on IPA and data driven similarity measures [4, 13]. Yet, all these models require a conventional Baum-Welch training for the HMM models with speech of every language, and as soon as a new language is included, this can affect the performance on all languages. For many practical systems, however, it would be more valuable to have one distinguished main language. An example is a voice operated car-navigation system. The user interacts most of the time in his native language, when he has to spell a telephone number, to enter an address and so on. However, there is also the need to offer multilingual recognition, as sometimes he might ask to drive to a foreign city or, more frequently, to select a song with a foreign title by voice command. For such a task, it is more important to offer as many languages as possible, instead of focusing on the usual goal of supporting a handful of selected languages and optimize their performance as much as possible. In our previous work [9], we showed that using a semi-continuous speech recognizer is an efficient way to recognize multiple languages with a limited number of parameters by using only one single set of Gaussians for all HMM models. In [10] we could show that this approach also helps for non-native speakers. However, such systems depend on the main language of the user, as this determines which Gaussians are considered in the recognition. This increases the training effort exponentially, as the HMMs have to be trained for many different sets of Gaussians. In this paper, we present a method to remove this additional training effort by projecting the distribution of an HMM state to another set of Gaussians. With this method, any language for which a monolingual recognizer was trained can be recognized with any set of Gaussians. The projection itself is based on an L2 distance between Gaussian Mixture Models (GMMs) as defined in [3].

The next section describes the distance between GMMs. The optimal projections section shows how a distribution is projected from one GMM to another GMM. After this, our experimental results are presented and a conclusion is drawn.

Distance between GMMs

The L2 distance between two probability distributions **A** and **B** defined on two different sets of Gaussians is defined by

$$D_{L2}(\mathbf{A},\mathbf{B}) = \int (\boldsymbol{\alpha}^T \mathbf{a}(\mathbf{x}) - \boldsymbol{\beta}^T \mathbf{b}(\mathbf{x}))^2 d\mathbf{x} \qquad (1)$$

$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the weight vectors of the the Gaussian vectors $\mathbf{a}$ and $\mathbf{b}$.

$$\boldsymbol{\alpha} = \begin{pmatrix} w_1^a \\ w_2^a \\ \vdots \\ w_n^a \end{pmatrix}, \mathbf{a}(\mathbf{x}) = \begin{pmatrix} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1^a, \boldsymbol{\Sigma}_1^a) \\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2^a, \boldsymbol{\Sigma}_2^a) \\ \vdots \\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_n^a \boldsymbol{\Sigma}_n^a) \end{pmatrix} \tag{2}$$

$$\boldsymbol{\beta} = \begin{pmatrix} w_1^b \\ w_2^b \\ \vdots \\ w_m^b \end{pmatrix}, \mathbf{b}(\mathbf{x}) = \begin{pmatrix} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1^b, \boldsymbol{\Sigma}_1^b) \\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2^b, \boldsymbol{\Sigma}_2^b) \\ \vdots \\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m^b, \boldsymbol{\Sigma}_m^b) \end{pmatrix} \tag{3}$$

The distance D_{L2} can be calculated as follows

$$\begin{aligned} D_{L2}(\mathbf{A},\mathbf{B}) &= \int \left(\boldsymbol{\alpha}^T\mathbf{a}(\mathbf{x}) - \boldsymbol{\beta}^T\mathbf{b}(\mathbf{x})\right)^2 d\mathbf{x} \\ &= \int [(\boldsymbol{\alpha}^T\mathbf{a}(\mathbf{x}))^2 \\ &\quad - 2\boldsymbol{\alpha}^T\mathbf{a}(\mathbf{x})\boldsymbol{\beta}^T\mathbf{b}(\mathbf{x}) \\ &\quad + (\boldsymbol{\beta}^T\mathbf{b}(\mathbf{x}))^2]\, d\mathbf{x} \\ &= \sum_i\sum_j \alpha_i\alpha_j \int a_i(\mathbf{x})a_j(\mathbf{x})\, d\mathbf{x} \\ &\quad - 2\sum_i\sum_j \alpha_i\beta_j \int a_i(\mathbf{x})b_j(\mathbf{x})\, d\mathbf{x} \\ &\quad + \sum_i\sum_j \beta_i\beta_j \int b_i(\mathbf{x})b_j(\mathbf{x})\, d\mathbf{x} \end{aligned} \tag{4}$$

with $a_i(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a)$ and $b_i(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)$. To solve this problem, the correlation $\int \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\, d\mathbf{x}$ between the Gaussians needs to be calculated. [7] state that

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) = c_c\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{5}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and the covariances of the Gaussians. The elements of the resulting Gaussian $c_c\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ are

$$\begin{aligned} c_c &= \mathcal{N}(\boldsymbol{\mu}_1; \boldsymbol{\mu}_2, (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}2)) \\ &= \frac{1}{\sqrt{det(2\pi(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2))}} \\ &\quad \cdot e^{\left[-1/2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right]} \end{aligned} \tag{6}$$

$$\boldsymbol{\mu}_c = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) \tag{7}$$

$$\boldsymbol{\Sigma}_c = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} \tag{8}$$

Thus

$$\begin{aligned} &\int \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\, d\mathbf{x} \\ &= \int c_c\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) d\mathbf{x} \\ &= c_c \underbrace{\int \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) d\mathbf{x}}_{=1} \\ &= c_c \end{aligned} \tag{9}$$

With this, all correlations between all Gaussians can be calculated and written in three matrices $\mathbf{M}^{AA}$, $\mathbf{M}^{AB}$ and $\mathbf{M}^{BB}$.

$$M_{ij}^{AA} = \int a_i(\mathbf{x})a_j(\mathbf{x})d\mathbf{x} \tag{10}$$

$$M_{ij}^{AB} = \int a_i(\mathbf{x})b_j(\mathbf{x})d\mathbf{x} \tag{11}$$

$$M_{ij}^{BB} = \int b_i(\mathbf{x})b_j(\mathbf{x})d\mathbf{x} \tag{12}$$

Hence Equation (4) can be written as

$$\begin{aligned} D_{L2}(\mathbf{A},\mathbf{B}) &= \sum_i\sum_j \alpha_i\alpha_j M_{ij}^{AA} \\ &\quad - 2\sum_i\sum_j \alpha_i\beta_j M_{ij}^{AB} \\ &\quad + \sum_i\sum_j \beta_i\beta_j M_{ij}^{BB} \\ &= \boldsymbol{\alpha}^T\mathbf{M}^{AA}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T\mathbf{M}^{AB}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{M}^{BB}\boldsymbol{\beta} \end{aligned} \tag{13}$$

Optimal Projections

In the previous section, a distance between two probability distributions $\mathbf{A}$ and $\mathbf{B}$ was defined. However, our goal is to map a probability distribution defined on one set of Gaussians to another set of Gaussians. In this case, $\mathbf{B}$ is completely defined, and a $\boldsymbol{\alpha}_{min}$ must be found that represents $\mathbf{B}$ as well as possible with the Gaussians in set $\mathbf{a}$. To find this $\boldsymbol{\alpha}_{min}$ we derive D_{L2} with respect to $\boldsymbol{\alpha}$:

$$\frac{\partial D_{L2}}{\partial \boldsymbol{\alpha}} = (\mathbf{M}^{AA} + \mathbf{M}^{AA^T})\boldsymbol{\alpha} - 2\mathbf{M}^{AB}\boldsymbol{\beta} \tag{14}$$

In order to find the minimum, we have to set the gradient to $\vec{\mathbf{0}} = (0, 0, \ldots, 0)^T$.

$$(\mathbf{M}^{AA} + \mathbf{M}^{AA^T})\boldsymbol{\alpha}_{min} - 2\mathbf{M}^{AB}\boldsymbol{\beta} = \vec{\mathbf{0}} \tag{15}$$

Solving this equation leads to the optimal weights $\boldsymbol{\alpha}_{min}$.

$$\boldsymbol{\alpha}_{min} = 2(\mathbf{M}^{AA} + \mathbf{M}^{AA^T})^{-1}\mathbf{M}^{AB}\boldsymbol{\beta} \tag{16}$$

This $\boldsymbol{\alpha}_{min}$ is a true minimum when the second derivative of D_{L2} is positive definite. The second derivative is $2M^{AA}$. M^{AA} is a correlation matrix, and therefore positive semidefinite. As long as none of the Gaussians is linearly dependent on the other Gaussians, this matrix is positive definite and therefore $\boldsymbol{\alpha}_{min}$ a true minimum.

However, this minimum is not a probability distribution, as the elements of $\boldsymbol{\alpha}_{min}$ do not sum to one, and there can be negative weights for Gaussians. In addition to the mathematical flaw, this poses problems for the decoding within a speech recognizer. The first problem is that our projection only considers one state distribution at a time. If $\sum_i \boldsymbol{\alpha}_i > 1$ for state1 and $\sum_i \boldsymbol{\alpha}_i < 1$ for state2, than state1 assigns in average a higher score to the observed feature vectors. This means that the comparability between the states is lost. The second problem is that negative weights can not be represented in log probabilities, which are frequently used in speech decoders. Therefore we need to find an $\boldsymbol{\alpha}_{min}$ that is a probability distribution for the successful application in a speech recognizer.

To enforce the sum equals one constraint, a Lagrange constraint can be added to the function. The new Lagrange function to minimize is:

$$\begin{aligned} L(\alpha, \lambda) &= \boldsymbol{\alpha}^T \mathbf{M}^{AA} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \mathbf{M}^{AB} \boldsymbol{\beta} \\ &+ \quad \boldsymbol{\beta}^T \mathbf{M}^{BB} \boldsymbol{\beta} + \lambda(\sum_i (\alpha_i) - 1) \qquad (17) \end{aligned}$$

with the additional Lagrange multiplier λ. Deriving this function gives

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = (\mathbf{M}^{AA} + (\mathbf{M}^{AA})^T)\boldsymbol{\alpha} - 2\mathbf{M}^{AB}\boldsymbol{\beta} + \begin{pmatrix} \lambda \\ \vdots \\ \lambda \end{pmatrix} \qquad (18)$$

$$\frac{\partial L}{\partial \lambda} = \sum_i (\alpha_i) - 1 \qquad (19)$$

The following gives one representation that shows both derivatives in closed form

$$\begin{aligned} \frac{\partial L}{\partial (\boldsymbol{\alpha}, \lambda)} &= \begin{pmatrix} \mathbf{M}^{AA} + (\mathbf{M}^{AA})^T & \vec{\mathbf{1}} \\ \vec{\mathbf{1}}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \lambda \end{pmatrix} \\ &- \begin{pmatrix} 2\mathbf{M}^{AB} & \vec{\mathbf{0}} \\ \vec{\mathbf{0}}^T & 1/\lambda \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta} \\ \lambda \end{pmatrix} \qquad (20) \end{aligned}$$

where $\vec{\mathbf{0}} = (0, 0, \ldots, 0)^T$ and $\vec{\mathbf{1}} = (1, 1, \ldots, 1)^T$.

Setting the derivative to $\vec{\mathbf{0}}$ and removing λ from the second matrix ($1/\lambda \times \lambda = 1 = 1 \times 1$) leads to

$$\begin{aligned} \begin{pmatrix} \boldsymbol{\alpha}_{min} \\ \lambda \end{pmatrix} &= \begin{pmatrix} \mathbf{M}^{AA} + (\mathbf{M}^{AA})^T & \vec{\mathbf{1}} \\ \vec{\mathbf{1}}^T & 0 \end{pmatrix}^{-1} \\ &\cdot \begin{pmatrix} 2\mathbf{M}^{AB} & \vec{\mathbf{0}} \\ \vec{\mathbf{0}}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta} \\ 1 \end{pmatrix} \qquad (21) \end{aligned}$$

This results in an $\boldsymbol{\alpha}$ vector that sums up to one. The second constraint of only positive weights for all Gaussians can be enforced with Karush Khun Tucker constraints [5]. These are basically a generalization of the Lagrange constraints and can work with inequalities by introducing slack variables $\mathbf{s}$ that transform every inequality in an equality, which can be solved as any Lagrange constraint. In the case here, an inequality constraint has to be introduced for every element of $\boldsymbol{\alpha}$. This gives the new function KKT for the distance between two distributions $\mathbf{A}$ and $\mathbf{B}$.

$$\begin{aligned} KKT(\boldsymbol{\alpha}, \lambda, \boldsymbol{\gamma}) &= \boldsymbol{\alpha}^T \mathbf{M}^{AA} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \mathbf{M}^{AB} \boldsymbol{\beta} + \boldsymbol{\beta}^T \mathbf{M}^{BB} \boldsymbol{\beta} \\ &+ \lambda(\sum_i (\alpha_i) - 1) + \sum_{i=1}^{n} \gamma_i(-\alpha_i + s_i^2) \end{aligned} \qquad (22)$$

with $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ and $\mathbf{s} = (s_1, s_2, \ldots, s_n)$.

The KKT function has always to give the same result as D_{L2}. This means either γ_i is zero, or $\alpha_i + s_i^2$ is zero. When α_i is zero, constraint i is said to be active, otherwise the constraint is inactive. If constraint i is active, γ_i is greater 0. To find the optimal solution, all possible combinations of active constraints and inactive constraints need to be evaluated, and one of these solutions will be the optimal solution that fits the constraints.

In practice it is not possible to check all the possible combinations for the optimal value. Similar problems have to be solved for Neural Networks [8, 1]. Basically, the idea is to perform a gradient descent on the optimization criterion and a gradient ascent on the equality constraint. [1] shows that a quadratic optimization problem that ignores negative values converges with gradient descent. In our case, the actual implementation was rather sensitive to good setting of update weights, as we have both equality and inequality constraints. Nevertheless, our sequential iterative optimization algorithm achieved with only three iterations an almost optimal projection that satisfied our constraints.

Experimental Setup

Our semi-continuous HMM speech recognizer uses 11 MFCCs with their first and second derivatives per frame and LDA for feature space transformation. Monolingual recognizers for English, French, Spanish, Italian and German are trained on 200 hours of Speecon data [2] with 1024 Gaussians in the codebook. The HMMs are context dependent and the codebook for each language is different. We have between 2000-3000 HMMs for each language. The language model is specified as a context free grammar.

Table 1 describes the test sets. The test sets are from proprietary in-car data. Each set consists of city names uttered by mative speakers. The number of different city names in our context free grammars is specified in the third column of Table 1.

Table 1: Description of the test set for each language

Testset	Language	Speech Items	Vocab.
GE_City	German	2005	2498
US_City	English	852	500
IT_City	Italian	2000	2000
FR_City	French	3308	2000
SP_City	Spanish	5143	3672

Experiments

In initial experiments, we experimented with using the projections that do not give probability distributions. However, our recognizer did not perform well with Gaussians weight vectors that did not sum to one and/or contained negative weights. Therefore, all experiments in this section only present results from our iterative approximation. This iterative approximation took about 400 seconds for 2000 HMM models. This is roughly the number of context dependent HMMs that our systems have for one language.

To verify that our approximation works, we compared the achieved L2 distances when we projected our monolingual English HMMs to the German codebook. The optimal projection without constraints obtained an overall distance of 4.08e-9. Our approximative projection still achieved a distance of 4.10e-9, only little worse.

Table 2 shows the results of systems that were projected to the German codebook with 1024 Gaussians. It can be seen, that the recognition is still working for all languages and that the recognition performance of the main language is not affected. However, the performance is significantly lower than a traditional training of HMMs on the German codebook. Thus minimizing the L2 distance is not giving a performance close to the maximum possible performance with a given set of Gaussians. In the future, we want to test other possible methods for projection HMM states from one codebook to another. Furthermore, we want to combine this new technique of projecting HMMs with our previous work of Multilingual Weighted Codebooks (MWCs) [10]. This should eliminate the most severe errors of our projection, as MWCs contain the most different Gaussians from many languages.

Table 2: Word Accuracies of HMMs created with our proposed projection

Testset	Language	Projection	Retraining
GE_City	German	84.1	84.1
US_City	English	55.5	65.6
IT_City	Italian	78.1	85.2
FR_City	French	59.3	68.7
SP_City	Spanish	71.2	88.3

Conclusion

In this paper we have motivated that it is beneficial for many Human Machine Interfaces to allow multilingual speech recognition without affecting the main language performance of the speech recognition system. We have thought about how to support the recognition of as many languages of possible with limited resources, and found a convenient answer in the traditional technique of semi-continuous speech recognition. The drawback of this solution is that each language has to be trained for every set of Gaussians, each of them determined by the currently supported main recognition language. An analysis of the problem showed that projections between Gaussian mixture feature spaces would be an elegant way of supporting recognition for many languages. Nevertheless, our experimental results show that the used projection, though mathematically optimal, is not optimal for the actual speech recognition performance. In the future, we expect to see improvements through combinations with our previous work, and are also researching for different ways how the projection can be achieved. Finally, we expect that our work will be of benefit to different areas of research, as GMMs are a widely used technique. While the proposed projection was not optimal for our goals, in other cases the L2 distance might be more closely related to the desired behavior of a system.

References

[1] M. Biehl, J. K. Anlauf, and W. Kinzel. Perceptron learning by constrained optimization: the AdaTron algorithm. In *Proc. ASI Summer Workshop Neurodynamics*, Clausthal, Germany, 1990.

[2] D. Iskra, B. Grosskopf, K. Marasek, H. van den Huevel, F. Diehl, and A. Kiessling. Speecon - speech databases for consumer devices: database specification and validation. In *Proc. LREC*, pages 329–333, Las Palmas de Gran Canaria, Spain, 2002.

[3] J. H. Jensen, D. P. W. Ellis, M.G. Christensen, and S. H. Jensen. Evaluation of distance measures between Gaussian mixture models of MFCCs. In *Proc. ISMIR*, pages 107–108, 2007.

[4] J. Koehler. Multilingual phone models for vocabulary-independent speech recognition tasks. *Speech Communication Journal*, 35(1-2):21–30, 2001.

[5] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proc. of 2nd Berkeley Symposium*, pages 481–492, Berkeley, California, 1951.

[6] P. Ladefoged. The revised international phonetic alphabet. *Language*, 66(3):550–552, 1990.

[7] K.B. Petersen and M.S. Pedersen. The matrix cookbook, 2008. `http://matrixcookbook.com`.

[8] J. C. Platt and A. H. Bar. Constrained differential optimization for neural networks. Technical report, Caltech, USA, 1988.

[9] M. Raab, R. Gruhn, and E. Nöth. Multilingual weighted codebooks. In *Proc. ICASSP*, pages 4257–4260, Las Vegas, USA, 2008.

[10] M. Raab, R. Gruhn, and E. Nöth. Multilingual weighted codebooks for non-native speech recognition. In *Proc. TSD*, pages 485–492, Brno, Czech Republic, 2008.

[11] T. Schultz and A. Waibel. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35:31–51, 2001.

[12] U. Uebler. Multilingual speech recognition in seven languages. *Speech Communication*, 35:53–69, 2001.

[13] Z. Wang, U. Topkara, T. Schultz, and A. Waibel. Towards universal speech recognition. In *ICMI*, pages 247–252, Pittsburgh, USA, 2002.