In *Proceedings of ICASSP*, volume 1, pages 639–642, New York, 1989.

[3] Biing-Hwang Juang. Maximum likelihood estimation for mixture multivariate stochastic observation of Markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[4] Biing-Hwang Juang and Lawrence R. Rabiner. The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:1639–1641, 1990.

[5] Teuvo Kohonen. Clustering, taxonomy, and topological maps of patterns. In *Proceedings of the 6th International Conference on Pattern Recognition*, volume 1, pages 114–128, Munich, Germany, 1982.

[6] Teuvo Kohonen. The Self-Organizing Map. In *Proceedings of the IEEE*, pages 1464–1480, 1990.

[7] Teuvo Kohonen, Jari Kangas, Jorma Laaksonen, and Kari Torkkola. LVQPAK: A program package for the correct application of learning vector quantization algorithms. In *Proceedings of the International Joint Conference on on Artificial Neural Networks*, Baltimore, June 1992.

[8] Kai-Fu Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(4):599–609, April 1990.

[9] Louis A. Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions on Information Theory*, 5:729–734, 1982.

[10] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of Fifth Berkeley Symposium on Math. Statist. and Prob.*, pages 281–297, 1967.

[11] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of ICASSP*, volume 1, pages 267–295, 1989.

[12] Kari Torkkola, Jari Kangas, Pekka Utela, Sami Kaski, Mikko Kokkonen, Mikko Kurimo, and Teuvo Kohonen. Status report of the finnish phonetic typewriter project. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 1, pages 771–776, Espoo,Finland, June 1991.

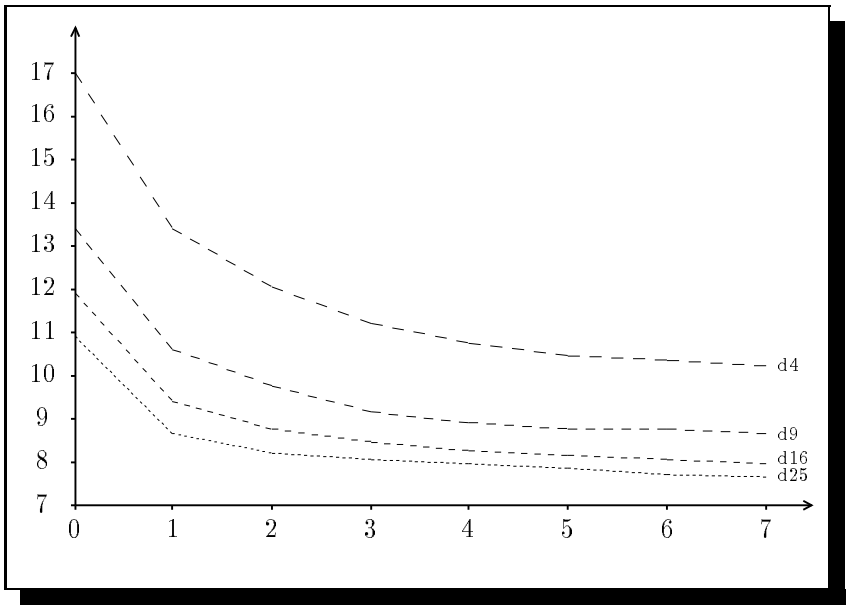

Figure 6: *Development of speech recognition error rate during Baum-Welch iteration with different mixture distribution sizes. The code "dnumber" indicates the number of multivariate Gaussian distributions used for each state. Initialization is in all cases done by Self-Organizing Maps.*

CONCLUSIONS

A new combined method for training CDHMMs is introduced. Self-Organized Maps and Learning Vector Quantization are used to select good initial parameter values for the approximative observation distributions of states. Improvements in the speed of convergence of the iterative Baum-Welch algorithm and in the performance in speech recognition are obtained. When models grow in size and complexity and more training data must be exploited the importance of initialization and fast convergence becomes even more evident.

References

[1] Herve Bourlard and Christian J. Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167–1178, December 1990.

[2] X.D. Huang and M.A. Jack. Unified techniques for vector quantization and hidden Markov modelling using semi-continuous models.

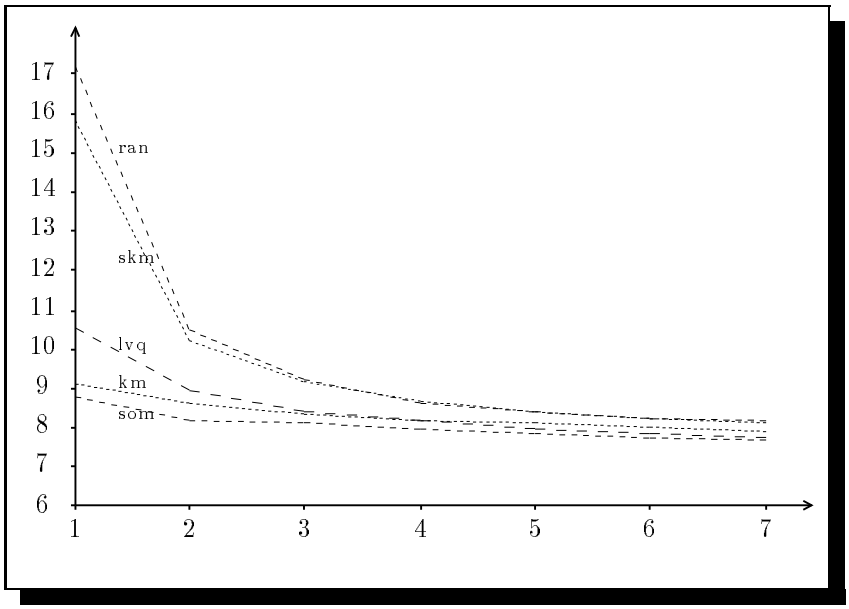

Figure 5: *Development of speech recognition error rate during Baum-Welch iteration with different initializations "ran", "km", "lvq" and "som". Error rate for segmental K-means training, which is more approximative and faster than B-W because it uses less data, is illustrated by the curve "skm". Number of iterations is shown in horizontal axis and percentage of recognition errors for independent test data in vertical axis. The curve "ran" corresponds the random initialization, "km" the K-means, "lvq" the LVQ3 and "som" the method of Self-Organizing Maps.*

probability density function of observations has an influence on the recognition rate development which is illustrated by figure 6.

The best obtained model gave an average 94.3% right phonemes for the tested speakers and an average error rate of 7.4% including changed, missing and extra phonemes in decoded phonetic descriptions.

Examination of the error rate curves in figure 5 reveal that a careful initalization saves many Baum-Welch iteration rounds. This is important because by adding a few rounds of SOMs (22s each) to the initialization phase (71s) spares several B-W rounds (260s each) which consume huge amount of memory. The segmental K-means algorithm is faster than Baum-Welch consuming a little more than 100s per iteration but requires quite many iterations before an acceptable recognition error rate is achieved. The execution times were measured in a Silicon Graphics Indigo workstation and should only be considered as qualitative.

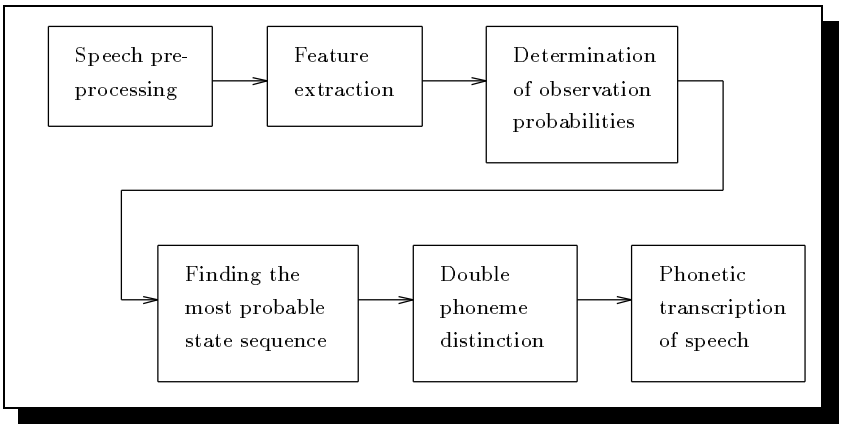

Figure 4: *Distinct phases of producing phonetic description of spoken finnish using hidden Markov models.*

changed, missing and extra phonemes in decoded phonetic transcriptions divided by the total number of phonemes.

Use of Self-Organizing Maps to give good initial values for the mixture distributions of the states in HMMs led to significantly faster convergence compared to the randomly chosen initial values (fig. 5). Also the final error rate dropped about 7 %. The segmental K-means algorithm which ought not to be influenced so much by poor initialization had only slightly better performance than Baum-Welch estimation with random initial values.

The Self-Organizing Maps were trained in this experiment as follows: Map size for each state was 5 times 5. Training data contains 5211 phonemes. Each phoneme sample was divided into 4 groups of feature vectors; one for each state in the HMM (see fig. 2). The feature vectors were then used in random order to update the corresponding originally random valued map. The training data was used 5 times during which the teaching gain was decreased monotonically from 0.2 to 0 and neighborhood radius from 3 to 0.

When LVQ algorithms were used for initialization the obtained recognition error rates were also lower and the convergence faster than with randomly chosen initial values. In this application of LVQ the vector to be updated corresponding to a training feature vector was selected by finding the closest Gaussian mean vector in the group of all mixture components in all state output density functions. For adjustments the learning laws LVQ1, LVQ2, LVQ3 [6] and the optimized learning rate OLVQ [7] were tried with almost equal recognition rates. In figure 5 the error rate development is illustrated for the LVQ3. The initialization for LVQ was done by finding a group of vectors which satisfy the K-nearest neighbor (KNN) criterion as suggested in [7].

Increasing the number of the Gaussian distributions to approximate the

initialization methods are profitable.

Clustering by SOM

In the task of fitting a combination of several simple parametric distributions to observed features the clustering approach is applicable. Each distribution component might then represent one cluster. The Self-Organizing Maps is an automatic clustering method which adaptively constructs a mapping from a high-dimensional vector space to a group of topologically connected cells or reference vectors [6]. Other algorithms can also be used for clustering, for example, the K-means algorithm [10], which is somehow similar to SOM, except that it doesn't conserve the topology. It produces more recognition errors than SOM (see fig. 5) after the same Baum-Welch training, however.

Improving discrimination by LVQ

In speech recognition, as in all pattern recognition, the objective is not to obtain extremely representative models, however, but to eliminate recognition errors. To reach this objective it is desirable that HMMs can produce a distinct difference in probabilities of wrong and right phoneme models using the obtained features. The Learning Vector Quantization algorithms [6] construct a set of reference vectors which minimizes the number of misclassifications in the training data. Tuning the initial values by LVQ provides one way to guide the Baum-Welch iteration towards more differentiative models.

EXPERIMENTS

The proposed modeling and training methods have been experimented with the speech recognition system in the Laboratory of Information and Computer Science at Helsinki University of Technology. The tests were performed for three male Finnish speakers. For each speaker, four repetitions of a set of 311 words were available.

The hidden Markov models of 20 Finnish phonemes were trained by extracting the phoneme samples from three word sets spoken by the same speaker. The fourth set was then used for testing. The system works like a phonetic typewriter writing the phonetic transcriptions of spoken Finnish words which are then compared with the correct transcriptions. The main phases of recognition are shown in figure 4.

The recognition performance for each differently trained model was determined by calculating the average of 12 recognition tests containing all 4 combinations of training and testing data for each of the 3 speaker. The resulting error percentage plotted in figures 5 and 6 is the sum of

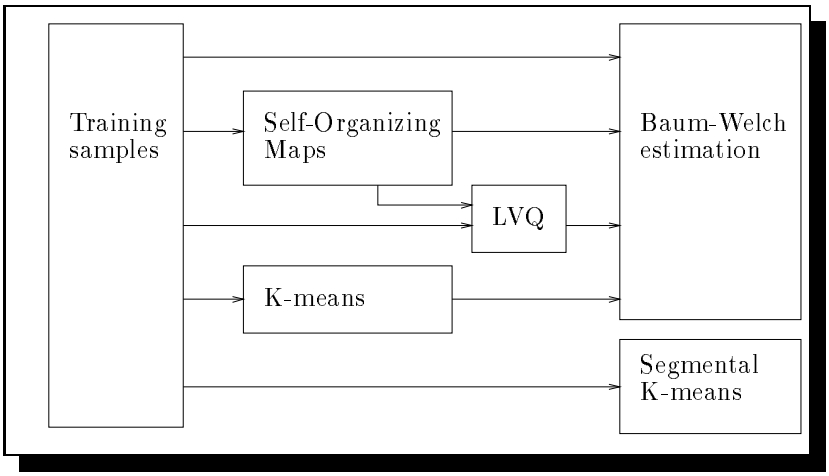

Figure 3: *Different training combinations for observation distributions.*

Increasing the number of mixture components makes the proper initial placement of the components nontrivial. Clustering of training samples would assure the best possible exploitation of all component distributions. The different training combinations experimented with are presented in figure 3.

The Baum-Welch reestimation is of course not the only method which can be used for the training of HMMs. Several interesting methods have been proposed in the last few years to direct the training objective towards differentiation or to reduce the amount of computations in the reestimation by different approximations. One method is the segmental K-means algorithm [4] which lightens computation by using only the most likely state sequence of each training sequence formed by Viterbi search with current parameter values for parameter reestimation. The other desired property of this algorithm is greater independence of parameter initialization.

INITIALIZATION METHODS

Random initial values

Choosing the initial values for the continuous observation distributions randomly is often adequate because the Baum-Welch iteration tends to converge relatively fast, at least in the case of simple mixture distributions. There is also no generally applicable optimal and well justified initialization method which would guarantee the best possible initial values for the Baum-Welch iteration. However, when large mixtures of high-dimensional feature vector distributions are required advanced

Keeping the covariance parameters same for each component of the mixture distribution allows to use considerably more components without increasing the computational load excessively. Also the assumption of the diagonality of the covariance matrix has the same effect. Proper estimation of covariance terms requires many representative feature vectors produced by each state and poor estimates can spoil the whole mixture probability density function.

To get a good representation of the observation density as well as an efficient discrimination between densities of different states the number of Gaussian density functions used in the mixture is about between 9 and 25 according to our experiments (fig. 6). With fewer components the observation density functions were maybe not representative enough because the recognition accuracies dropped clearly when reducing components from 9 to 4. Use of more than 25 Gaussians (e.g. 36) didn't improve the results. To be able to estimate and to use this kind of huge mixtures certain generalizations have to be made with the parametric distributions.

TRAINING THE MODEL PARAMETERS

In this paper we estimate the parameters of CDHMMs by the iterative Baum-Welch maximum likelihood reestimation algorithm (see e.g. [11]). This algorithm aims to maximize the likelihood of the training data having been generated by the model. To model accurately the real observations it is required that the successive observations of the signal are independent and that the true distribution belongs to the group of models being searched. Despite of the fact that in speech recognition these assumptions are not actually valid very good results have been reported in the literature. These results verify the power and robustness of the Baum-Welch reestimation procedure.

The Baum-Welch reestimation has, however, some practical drawbacks. In real applications the amount of training data required to obtain accurate models is relatively large and makes reestimation cycles computationally heavy and memory consuming. When longer feature vectors and increased number of mixture components are used, the increased distribution complexity seems to require more iteration cycles before the models can be succesfully applied for speech recognition purposes.

It is possible to reduce the number of iterations by speeding up the convergence, if the reestimation is started from suitable initial values. A suitable iteration amount can be determined by checking the recognition accuracy after each iteration. When the error rate is no longer declining or is declining very slowly there is no point to continue training. With complicated models it is normal that different initial values result in different models that have also slightly varying recognition performances. An iteration can (in average) be forced to lead to better models by using better initialization (see fig. 5).

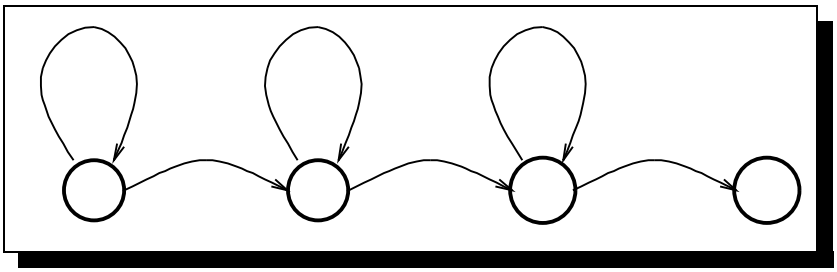

Figure 2: *The Markov model of a phoneme is 4-state uni-directional chain. Each state has its own continuous observation and discrete transition distributions.*

```
>>>>>>>>JJJJJJJJJJOOOOOOOOOOOOOOOOHHHHHHHHHHHHDDDDDDDDDDOOOO
00111123000111122230000111222222301111111112300122222230001

OOOOOOOSSSSSSSSSSSTTTTTTTTTTTTTTAAAAAAAA<<<<<<<<<<<<<<<<<<
12222230012222222230011111111112301111223011111111111222223
```

Table 1: *Modeling the sequence of feature vectors of figure 1 by HMMs. This is the most probable path through states found by Viterbi search that could have generated the observations. The number (0,1,2 or 3) below each phoneme label is the number of the current state in Markov model of that phoneme. The silence preceeding the word is indicated by label > and the following by <.*

raised sine [12]. Rms-value of the signal is concatenated to the vector.

The observations in one state of a phoneme-model HMM (see figure 2 and table 1) may have very complex probability density functions. This fact is due to the large variability of phonemes in different contexts. For example, fitting one multivariate Gaussian distribution produces normally poor results. If the number of models is allowed to increase, it is possible to take the context into account by defining separate models for different contexts [8]. A nonparametric model (for example an artificial neural network) can also be used to assign a probability value to each observed feature vector using characteristics adapted from given feature examples (e.g. [1]). To conserve the continuous properties of the probability density function and in the same time be able to adjust the probability distribution to complex structures, linear combinations of several simple parametric distributions (suchs as multivariate Gaussians) can be applied.

According to our observations better models can be obtained with less computation by using a combination of many similarly shaped Gaussians than a few individually shaped. Another commonly used way to reduce the number of estimated parameters are the tied mixture HMMs (e.g. [2]).

probability distribution is then estimated for each state. Combinations of these two approaches have also been suggested, for example, the semi-continuous HMMs [2]

Vector quantization reduces vastly the information content of the speech signal. Some essential information for the classification task can be lost in the process, however. On the other hand, if the quantization codebook is trained using the LVQ algorithms [6] the reference vectors can be selected optimally in the sense of maximizing the phoneme discrimination ability. Increased differentiation properties of observation probabilities have resulted in excellent recognition accuracies, for example, in the Finnish phonetic typewriter project [12].

A problem with CDHMMs in real applications is that when complex distributions are modeled accurately both the number of parameters and the computational work increase enormously. In this work, the initial estimates are formed by fast neural network based methods such as Self-Organizing Maps [5] and Learning Vector Quantization [6]. We suggest to speed up the training phase and guide it towards best solutions by using good initial esimates for the model parameters.

APPROXIMATION OF THE DISTRIBUTION OF OBSERVATIONS

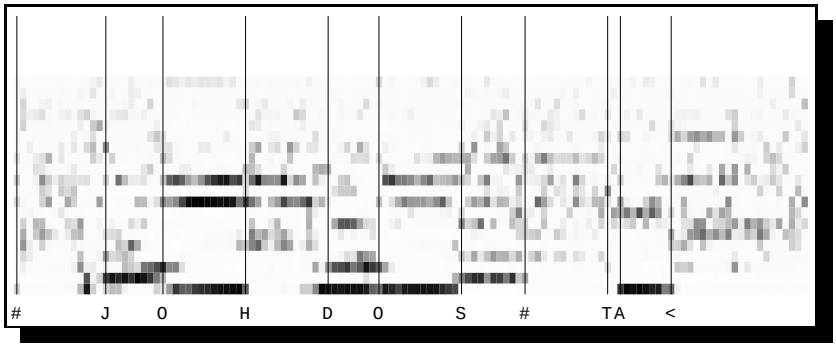

Figure 1: *An example of cepstral feature vectors of a spoken Finnish word "johdosta". The vectors are represented by vertical stripes in which the value of each component is illustrated by the gray level of corresponding small square. The starting point of each phoneme is marked by a vertical line, and the phoneme label is written below it. # denotes silence and < the end of the word.*

The features used in our speech recognition system are short-time mel-scale cepstral coefficients computed every 10 ms from a 20 ms window (figure 1). The feature vector, that is considered as an observation of the speech signal, contains 20 cepstral coefficients weighted (liftered) with

TRAINING CONTINUOUS DENSITY HIDDEN MARKOV MODELS IN ASSOCIATION WITH SELF-ORGANIZING MAPS AND LVQ

Mikko Kurimo†and Kari Torkkola‡
†Helsinki University of Technology
Laboratory of Information and Computer Science
Rakentajanaukio 2 C, SF-02150, FINLAND
tel: +358 0 451 3266, fax: +358 0 451 3277
email: mikko.kurimo@hut.fi
‡IDIAP, CP 609, CH-1920 Martigny, SWITZERLAND

Abstract. We propose a novel initialization method for continuous observation density hidden Markov models (CDHMMs) that is based on Self-Organizing Maps (SOMs) and Learning Vector Quantization (LVQ) [6]. Our framework is to transcribe speech into phoneme sequences using CDHMMs as phoneme models. When numerous mixtures of, for example, Gaussian density functions are used to model the observation distributions of CDHMMs, good initial values are necessary in order to the Baum-Welch estimation to converge satisfactorily. We experiment with constructing rapidly good initial values by SOMs, and with enhancing the discriminatory power of the phoneme models by adaptively training the state output distributions by using the LVQ-algorithm. Experiments indicate that an improvement to the pure Baum-Welch and the segmental K-means procedures can be obtained using the proposed method.

INTRODUCTION

HMMs are currently one of the most used methods in speech recognition. This method offers an efficient way to model a dynamically varying speech signal by dividing it into a chain of successive states each having different statistical properties (see fig. 2). Each state in the HMM acts like a Markov source emitting signal which is then observed. Due to relatively slow speed of state changes the observations of the signal can be considered as wide sense stationary during each state.

There are two major variants in handling the observation distributions. CDHMMs model the observed feature vectors directly by parametric probability density functions. Another way is to apply vector quantization to map the feature vectors to a discrete set of symbols whose