Hybrid training method for tied mixture density hidden Markov models using Learning Vector Quantization and Viterbi estimation

Mikko Kurimo
Helsinki University of Technology
Neural Networks Research Centre
Rakentajanaukio 2 C, FIN-02150, ESPOO, FINLAND
tel: +358 0 451 3266, fax: +358 0 451 3277
email: mikko.kurimo@hut.fi

Abstract. In this work the output density functions of hidden Markov models are phoneme-wise tied mixture Gaussians. For training these tied mixture density HMMs, modified versions of the Viterbi training and LVQ based corrective tuning are described. The initialization of the mean vectors of the mixture Gaussians is performed by first composing small Self-Organizing Maps representing each phoneme and then combining them to a single large codebook to be trained by Learning Vector Quantization (LVQ). The experiments on the proposed training methods are accomplished using a speech recognition system for Finnish phoneme sequences. Comparing to the corresponding continuous density and semi-continuous HMMs in [9] and [8] in the respect of the number of parameters, the recognition time and the average error rate, the performance of the phoneme-wise tied mixture HMMs is superior.

INTRODUCTION

Hidden Markov models are widely used in automatic speech recognition as phoneme models to combine the modeling of stationary stochastic processes producing observable short-time features and the temporal relationships between these processes. The temporal model in HMMs relies on a relatively simple structure of successive states and a probabilistic model of their mutual transitions.

The modeling of the stochastic observation processes associated with the states of HMMs is based on estimation of the probability density function of the short-time observations in each state. Several different approaches have been proposed to represent these output probabilities ranging from the estimation of parameters of multivariate Gaussian density [1] to the construction of multilayer perceptrons [3] or LVQ codebooks [5].

A common model category for the observation densities is the mixture density functions normally accomplished as a linear combination in a set of Gaussian densities. If the mixture densities are tied between all the HMM states the models are often called semi-continuous HMMs [4].

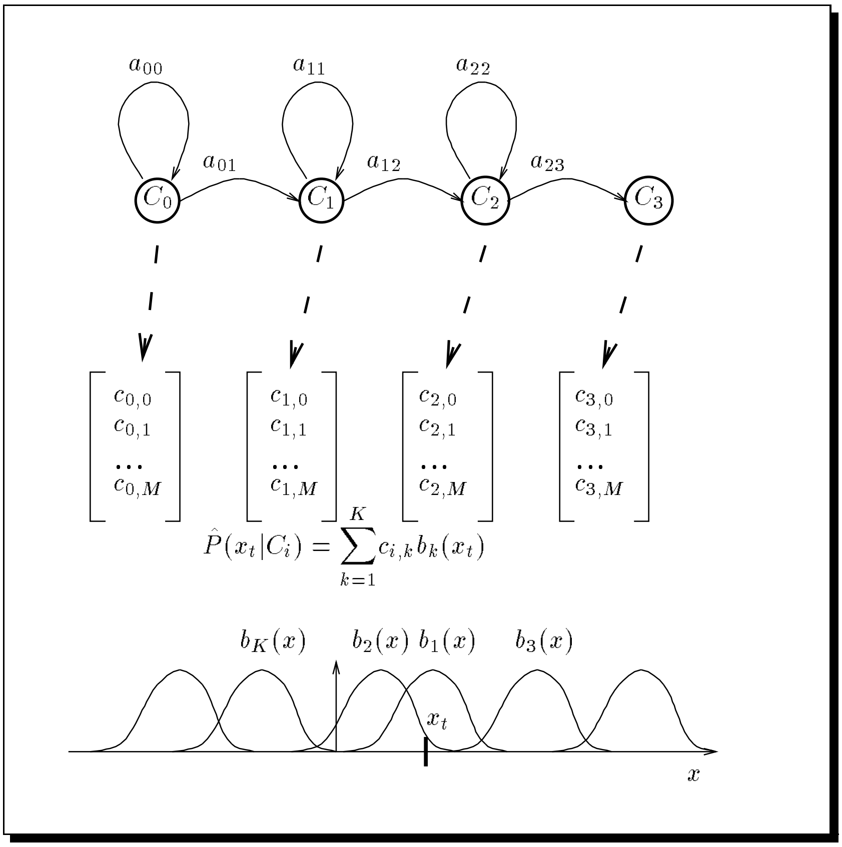

Figure 1: The output probability of state C_i at time t is computed using a tied mixture density function for K nearest mixture densities. The K nearest of all M mixture densities to the current observation x_t are indicated by indexes $k = 1, \cdots, K$. The mixtures are tied for states representing the same phoneme. The HMM is completely defined by the set of transition probabilities a_{ij}, mixture densities $b_k(x)$ and mixture weights $c_{i,k}$.

In this work the tying of the mixture densities is applied in a novel way so that the states belonging to the same HMM (Fig. 1), i. e. representing the same phoneme, use a common codebook of Gaussian densities. Thus there are as many sets of Gaussians as there are HMMs, which is a kind of intermediate for continuous density HMMs (different set for each state) and semi-continuous HMMs (only one large set of Gaussians).

The reason for the phoneme-wise tied Gaussian codebooks is to balance between a vast number of mixture mean vectors and covariances, like in large continuous density HMMs (CDHMMs), and an excessive amount of mixture weights, like in large semi-continuous HMMs (SCHMMs) (Table 1). From the CDHMM point of view, having common Gaussians for states of the same phoneme seems to be an appealing approximation, since the output densities of successive states are often highly overlapping. From the SCHMM point of view, this means that the normally quite a large set of very small weight values can be reduced, because most of the large weights are often nicely localized around Gaussians resembling to one phoneme [8].

LVQ FOR TIED MIXTURE DENSITIES

The Learning Vector Quantization (LVQ) [6], [7] methods are applied in the present paper to increase the discrimination between the phoneme models. This is a very important objective, since the primary interest in phoneme recognition is to find out the best-matching phonemes instead of estimating the correct probabilities for different phonemes.

In [10] the LVQ methods were applied to provide a discriminative initialization for the Gaussian mean vectors of CDHMMs leading to good recognition results with only a few iterations of the actual HMM parameter estimation performed by Baum-Welch reestimation. In [8] the same ideas were brought to the SCHMMs causing a significant drop in the average recognition error rates.

In the present paper with phoneme-wise tied mixture densities, the LVQ training is similar to the normal SCHMMs [8], except that the initial codebook for LVQ is combined from several smaller codebooks first prepared for each phoneme separately. After LVQ, the codebook is again split among the HMMs. The advantage of this combine-and-split procedure is that in contrast to the labeling method proposed in [8], the satisfactory representation of each phoneme can be guaranteed in addition to the enhanced discrimination between phonemes.

The small codebooks including only representatives of one phoneme can be efficiently trained by Self-Organizing Maps to capture the most essential features of each phoneme and provide a good basis for LVQ. The same initialization procedure could be successfully applied also to the mixture densities in CDHMMs by setting identical mean vectors for the output densities of states of one HMM. A separately trained small codebook of one phoneme is presented in Figure 2 as well as the same codebook after LVQ.

By examining carefully the differences between the corresponding cepstra in the upper and in the lower part of Figure 2 the development of the codebook can be recognized. The upper part is a small SOM (6x4 units) for phoneme /A/ trained in two phases. The first is a short ordering phase with large, but quickly reducing neighborhood radius 5 and learning rate 0.2. The second is a longer specialization phase starting with radius 2 and rate 0.02 reduced to 1 and 0, respectively. The result does not provide very low quantization error, but instead gives a rough representation of the pdf of the corresponding feature space. As an initialization, this presents a smooth two-dimensional surface fitted to the set of training samples with the borders more folded than the center.

The lower part of Figure 2 includes the same codebook vectors, but

Figure 2: The Gaussian mean vectors of short-time cepstral features for the tied mixture HMM of phoneme /A/ in the different training phases. First (top), after initialization by SOM, and second (bottom) after the LVQ training together with the other phoneme codebooks.

after the combine-and-split procedure and LVQ training. First, all the small SOMs, each representing one phoneme, are concatenated into one large LVQ codebook. The LVQ is started by a short training using the Optimized learning rate LVQ (OLVQ1) [7] followed by a longer training using LVQ3 [6]. The codebook vectors are divided into the same groups as in the original SOMs and each of these groups is converted to the set of mean vectors of the mixture Gaussian densities. Comparing to the upper part of Figure 2, the cepstra are now more distinct, providing a smaller quantization error and, which is more important, a better discrimination between the codebooks of other phonemes. In spite of the changes, some

Training Phase	Phoneme /A/ only			Whole data set	
	Qerror%	MMdist	Cacc%	Cacc%	Rerror%
SOM	17	3	85	61	24
+LVQ	13	13	87	66	8
+Viterbi	13	14	87	66	7
+Tuning	13	13	87	66	7
SOM	17	3	85	61	21
+Viterbi	13	14	87	66	8
+Tuning	15	14	77	61	7
KNN	19	17	83	53	34
+LVQ	16	8	94	49	55
+Viterbi	13	14	88	65	8
+Tuning	15	16	73	60	8
KM	17	10	80	57	18
+Viterbi	14	15	82	65	7
+Tuning	18	20	54	58	7

Table 1: Various statistics in different training phases for one data set (one speaker, 311 words). Qerror% is the average quantization error of the samples. MMdist is the median of distances from each codebook vector to its nearest neighbor. Cacc% is classification accuracy of sample vectors. Rerror% is the phoneme recognition error rate by corresponding tied mixture density HMM. KM is Kmeans clustering and KNN is the method to select the prototypes so that the majority of the K (K=5) nearest data points refer to correct classes.

structure is still visible giving smoothness to the codebook.

In Table 1 some characteristics of codebooks, i.e. the sets of Gaussian mean vectors, are shown numerically. In contrary to the statistically more covering experiments presented at the end of this paper, the values of Table 1 are not average rates but results from single experiments (one speaker, 311 words) and thus insignificant by themselves but only reflecting some general trends.

In Table 1 the characteristics of codebooks during four different training combinations 3–4 distinct phases in each have been analyzed. (The codebooks of phoneme /A/ of the two topmost rows were illustrated in Figure 2.) Kmeans clustering has been used as a substitution of SOM for minimizing quantization error without creating any structure. KNN refers to the basic initialization method for LVQ [7], which creates the codebook by directly selecting valid prototype vectors among the training samples.

The average quantization error reflects, how tightly the codebook is bound to the training samples of that phoneme. Generally, the Viterbi training, being based on the maximum likelihood principle, provides small quantization error. The corrective tuning, while increasing the differentiation between models, tends to increase the quantization error.

The median of the nearest neighbor distances is a measure of the general density of the codebook vectors, i. e. how close they are to each other. This measure shows quantitatively the phenomena observable from Figure 2 that SOM produces tight codebooks avoiding outliers.

The classification accuracy of the training samples simply shows how many of the samples will get the right lable in nearest neighbor classification without any context information. The low total accuracy is due the samples taken from the transition areas between phonemes and from certain plosives that are practically indistinguishable without wider context information.

Because the phoneme recognition error rate is the only measure here that takes account the temporal structure of HMMs, the Viterbi training, by enhancing directly the temporal model, naturally introduces a significant drop of errors. However, the LVQ, if suitably initialized, seems to provide mean vectors that, when combined with simple temporal model, are able to produce comparable error rates. Apparently, the short-time classification accuracy is not directly proportional to the phoneme recognition error rate.

VITERBI TRAINING AND CORRECTIVE TUNING

After the LVQ initialization, the next task in the estimation of the tied mixture density HMMs is the determination of the mixture weights and state transition probabilities. The initial values for the mixture weights can be simply assigned by finding the nearest Gaussians for a set of training vectors and computing the portion of hits for each Gaussian. If no pre-segmentation is available, the training samples can be divided into parts of equal lengths for this initialization. The actual training is accomplished by a version of Viterbi training, i. e. using an initial model to recognize the training words and to segment the sequence of observations into the parts corresponding to each HMM state. Parameter values for each state are then updated using the segmented feature vectors. For example, the resulting mixture weights of the different states representing phoneme /A/ with a tied Gaussian codebook of 24 mixtures are shown in Figure 3.

In Figure 3 the weights in each state are displayed as a surface over the mixtures. It is easy to notice that although some mixtures are used by several states, the main trend is that the center of greatest activation, i. e. largest weights, move from one corner of the codebook to the other.

The Viterbi search for the best path can be constrained on the available information about the phonemes in the training words to produce as correct path as possible [9] to reduce the number of required iterations. That information can include, for example, some pre-segmentation by other models or by hand. The search can be also made more efficiently by approximating the output density probabilities of the tied mixture density functions by using only a couple of the nearest mixtures to the

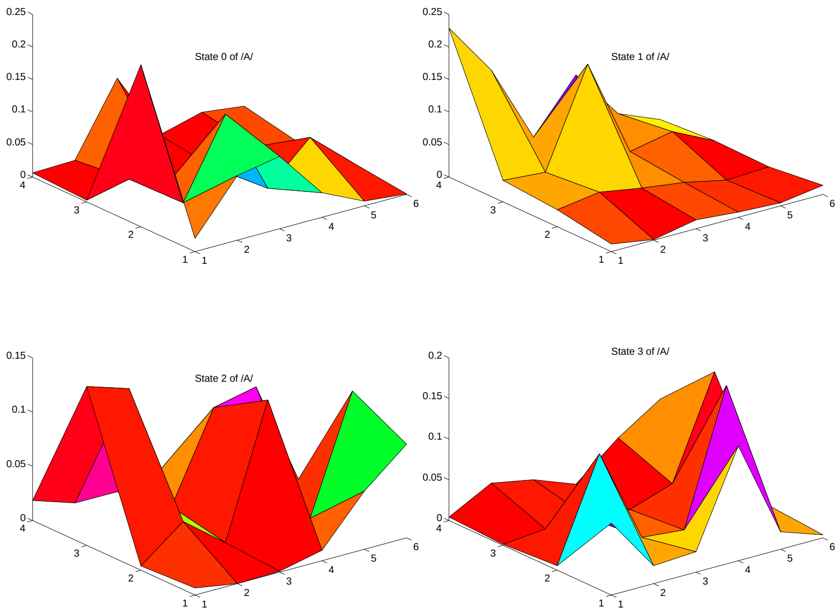

Figure 3: The mixture weights for the Gaussian mean vectors in the four states of tied mixture density HMM of phoneme /A/. The weights are grouped into a 6x4 array corresponding to the dimensions of the Self-Organizing Map used in the initialization (Figure 2).

current observations [2],[9].

To enhance the recognition ability of the models, the LVQ-based corrective tuning methods [9] can be applied to the tied mixture density HMMs of the current paper. These methods are approximative probabilistic descent algorithms tuning the HMMs to produce correct transcriptions by gradually decreasing modifications, if some phonemes in the training words get misclassified. Other methods with some common characteristics are introduced in, e. g. [11] and [12]. The main idea is to use the current models to find misrecognized words and the misrecognized phonemes in them. The incorrect part of the state sequence is then inspected state by state tuning the closest Gaussians that would give the correct result closer and the nearest incorrect ones away [9].

EXPERIMENTS

The experiments involve testing the phoneme recognition accuracy for four sets of 311 Finnish words uttered by three different speakers. By leaving one set at a time for testing totally 12 independent runs are obtained and their average error rates are used in comparison of the

version	error%	iterations
SOM	5.7	1000+10000
SOM+LVQ2	5.7	1000+10000+10000
KM	5.8	200 (rounds)
LVQ3	5.8	5000+50000
SOM*	6.6	1000+10000
SOM+LVQ3	5.7	1000+10000+5000+50000
SOM+LVQ3+Tuning	5.6	+5 times all words

Table 2: Average recognition error rates for variations of training combinations used for the Gaussian mean vectors for 5-state phonetically tied mixture HMMs. KM is the Kmeans algorithm, where 200 is the maximal number of rounds of the all training data for each phoneme. SOM* a reference experiment with SOM of zero neighborhood. SOMs are iterated in two phases (shorter and longer) for each phoneme. LVQ is iterated for all phonemes at the same time and a short OLVQ1 phase precedes the LVQ3 training. The corrective tuning is applied by presenting one word at a time and only misrecognized phonemes cause modifications.

Type of HMM	Number of mixtures	Parameters ($x10^3$)		Recognition	
		weights	means	time/word	error%
CDHMM	4	0.4	9	0.5	7.9
PWMHMM	24	3	11	0.7	6.9
SCHMM	494	54	10	1.8	8.1
CDHMM	24	3	55	2.2	5.8
PWMHMM	70	8	32	1.5	5.7

Table 3: Some comparisons between different continuous density HMM structures. The abbreviation PWMHMM refers to the phoneme-wise tied mixture density HMMs proposed in this paper. The CDHMM and SCHMM experiments, reported in earlier papers of the author, use the same speech database and basicly similar training phases as set up here for PWMHMM experiments. No corrective tuning has been used in these experiments.

training methods. The error rates are defined by the sum of missing, changed and extra phonemes divided by the correct sum of phonemes.

The phoneme recognition experiments are performed using the speech recognition system of the Laboratory of Information and Computer Science of Helsinki University of Technology [13]. 20 dimensional cepstral feature vectors concatenated with the energy of the signal are used as the short-time acoustical features. New feature vector is computed every 10 ms using 20 ms signal window.

5 states left-to-right HMMs with no skips are trained by the described

methods for 20 common finnish phonemes and for the silences occurring between the words. The differences of the average error rates in Table 2 are so small that, for example, the Matched Pairs test does not give any significant statistical differences between the various experimented training combinations, except that the experiment of SOM with zero neighborhood is worse than the others.

When comparing the performance of corresponding training combination between different continuous density HMMs the Table 3 reveals that the phoneme-wise tied HMMs provide clearly the most appealing configurations, when the number of parameters, the recognition time and the error rate are compared. The recognition times per word are computed as the average of 311 different finnish words and do not include the preprocessing which is same for each model. The experiments for CDHMMs and SCHMMs were explained in [9] and [8], but the construction of the experiments and the training algorithms are basicly the same and thus the results are comparable.

CONCLUSIONS

A new method to group the Gaussian codebooks for tied mixture density HMMs so that the mixtures are phoneme-wise tied is presented. The aim is to create an intermediate solution between the continuous density and semi-continuous HMMs gathering the best characteristics of both approaches. A hybrid training scheme that tries to combine the discrimination powers of LVQ and the presentation ability of Viterbi training is described for the new models. Some numerical and visual analysis of the created phoneme-wise codebooks of the Gaussian mean vectors are carried out as well as a comparison of different combinations of training algorithms. The performances of the best combinations are not significantly different but when comparing to the previous, corresponding experiments with continuous density and semi-continuous HMMs, significant improvements are achieved in respect of the number of parameters, the recognition time and the average error rates.

REFERENCES

[1] L. Bahl, F. Jelinek, and R. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983.

[2] Jerome Bellegarda and David Nahamoo. Tied mixture continuous parameter modeling for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(12):2033–2045, 1990.

[3] 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.

[4] X.D. Huang and M.A. Jack. Semi-continuous hidden Markov models for speech signals. *Computer Speech and Language*, 3, 1989.

[5] H. Iwamida, S. Katagiri, E. McDermott, and Y. Tohkura. A hybrid speech recognition system using HMMs with an LVQ-trained codebook. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 489–492, 1990.

[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. LVQ_PAK: A program package for the correct application of learning vector quantization algorithms. In *Proceedings of the International Joint Conference on Neural networks (IJCNN)*, Baltimore, June 1992.

[8] Mikko Kurimo. Using LVQ to enhance semi-continuous hidden Markov models for phonemes. In *Proceedings of 3rd European Conference on Speech Communication and Technology*, volume 3, pages 1731–1734, Berlin, Germany, September 1993.

[9] Mikko Kurimo. Corrective tuning by applying LVQ for continuous density and semi-continuous markov models. In *Proceedings of International Symposium on Speech, Image Processing and Neural Networks*, volume 2, pages 718–721, Hong Kong, April 1994.

[10] Mikko Kurimo and Kari Torkkola. Training continuous density hidden Markov models in association with self-organizing maps and LVQ. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 174–183, Copenhagen, Denmark, August 1992.

[11] Shinobu Mizuta and Kunio Nakajima. An optimal discriminative training method for continuous mixture density HMMs. In *Proceedings of the International Conference on Spoken Language Processing*, volume 1, pages 245–248, Kobe,Japan, November 1990.

[12] David Rainton and Shigeki Sagayama. Minimum error classification training of HMMs - implementation details and experimental results. *J.Acoust.Soc.Jpn*, 13(6):379–387, 1992.

[13] 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 ICANN*, volume 1, pages 771–776, Espoo,Finland, June 1991.