USING LVQ TO ENHANCE SEMI-CONTINUOUS HIDDEN MARKOV MODELS FOR PHONEMES

Mikko Kurimo

Helsinki University of Technology
Laboratory of Information and Computer Science
Rakentajanaukio 2 C, SF-02150, Espoo, FINLAND
email: mikko.kurimo@hut.fi

ABSTRACT

Experiments are made to enhance the discrimination ability of the SCHMMs by applying Learning Vector Quantization. The SCHMMs are used for the modeling of phonemes in a speaker-dependent speech recognition application to create the phonetic transcriptions of spoken utterances. The probability density functions for the cepstral feature vectors produced in each state of each model are modeled by mixtures of multivariate Gaussian density functions. The mean vectors of the Gaussian densities are chosen by clustering the feature vectors of the training samples by using the Self-Organizing Map (SOM). Then the Gaussians are modified to correspond better to the Bayesian decision surfaces between phonemes by tuning the mean vectors by the LVQ. The experiments indicate that by this careful placement of the mean vectors the recognition error rates for the SCHMMs decrease significantly. LVQ algorithms can also be successfully applied after Baum-Welch or Viterbi training to slightly modify the Gaussians using training samples which would otherwise be incorrectly recognized. This kind of error corrective tuning drops out some of the recognition errors also in the test data.

Keywords: *HMMs, LVQ, SOM, semi-continuous*

1 INTRODUCTION

The SCHMM [2] is actually a special form of the continuous density hidden Markov model where the set of Gaussian density functions used in the output mixture densities is shared between all the states in all Markov models. The SCHMM can be also regarded as a kind of intermediate between the discrete vector quantization based methods and the normal CDHMM.

The similar maximum likelihood training as used for the CDHMMs is also applicable to the SCHMMs and it can be expected that the presentation of discriminative training could also improve the performance of SCHMMs as well as that of CDHMMs like, for example, in [9].

The discriminative training method for CDHMMs derived in [4] (Generalized Probabilistic Descent, GPD) incorporates smooth classification error functions into the model optimization by gradient search. The gradient calculations for the modifications of the mixture density functions are, however, very time consuming and several simplifications are necessary for the practical implementations. In [6] it is suggested that the general idea behind the simplified GPD training is equivalent to the performing of LVQ2 [5] between the input and the closest reference pattern on best path found by dynamic programming. A corrective training algorithm resembling the LVQ2 is also presented in [9] although it suggests modification of the reference vectors even when the training sample is only nearly misrecognized.

The purpose of this paper is to study experimentally the LVQ training applied to the Gaussian mean vectors in SCHMMs both before the estimation of mixture weights and state transition probabilities and after that.

2 SCHMM STRUCTURE

The Markov models used in this paper are unidirectional chains of three to five states, which are assumed to produce observations following a certain continuous probability density function. The transitions between states occur according to a discrete set of transition probabilities. The transitions with a positive probability are either jumps to the next state in the chain or loops back to the current state (see figure 1).

The probability density functions for the cepstral feature vectors produced in each state of each model are modeled by mixtures of multivariate Gaussian density functions. The probability that state i can produce an observation O_t is then calculated by

$$P(O_t|C_i) = \sum_{m=1}^{M} c_{im} b_m(O_t) \quad , \tag{1}$$

where the components $b_m(O_t)$ are, for example K dimensional multivariate Gaussian densities

$$b_m(O_t) = \frac{1}{(2\pi)^{K/2}|\Sigma_m|^{1/2}} e^{-\frac{1}{2}(O_t-\mu_m)^T \Sigma_m^{-1}(O_t-\mu_m)} \tag{2}$$

and the weight factors $c_{im} < 1$ are positive real numbers which sum to 1 for each state i. The mixture coefficients are unique for each state, but the collection of M Gaussian densities is tied between the states in order to achieve

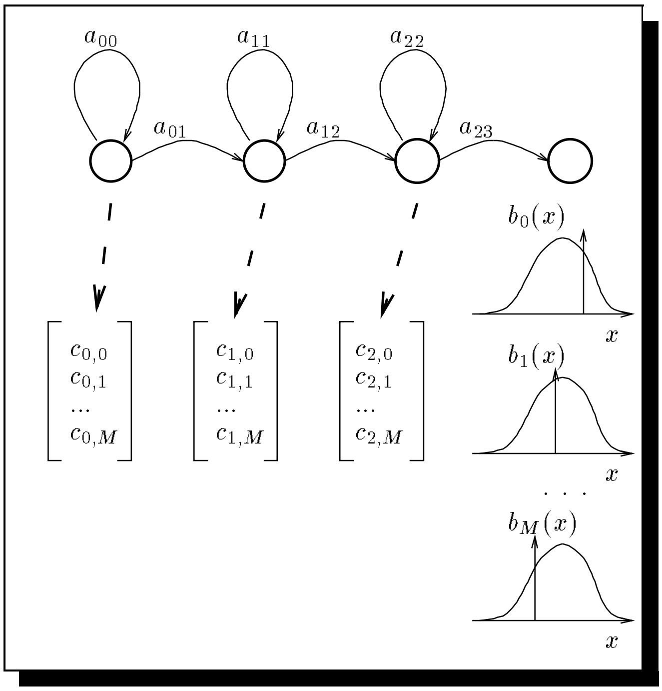
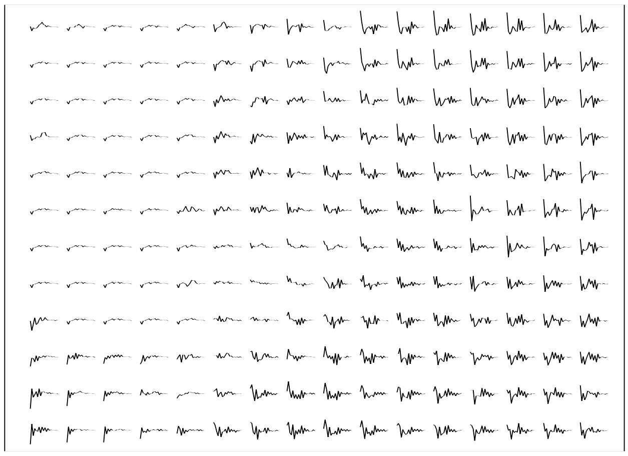


Figure 1: *The Markov model of a phoneme is 4-state unidirectional chain. Each state has a discrete transition distribution a_{ij} and its own set of coefficients c_{im} to produce its observation density function as a mixture from the global set of Gaussians $b_m(x)$. The observations x are projected to scalars here only for illustrative purposes.*



<	<	<	<	<	<	<	<	>	U	U	O	O	O	A	O
<	<	<	<	<	<	<	<	K	U	U	O	O	O	A	A
<	<	<	<	<	<	H	<	V	M	U	L	L	A	A	A
<	<	<	<	<	<	<	<	V	M	L	L	L	A	A	A
<	<	<	<	<	<	<	R	M	M	N	U	A	A	A	A
<	<	<	<	<	<	<	<	V	N	N	N	R	O	A	A
<	<	<	<	<	<	<	V	>	N	N	N	<	R	A	A
<	<	<	<	<	<	V	D	L	L	N	N	R	<	Ä	Ä
K	<	<	<	<	<	<	J	L	L	Y	Y	E	E	Ä	Ä
K	K	>	>	H	<	<	J	J	I	Y	I	E	E	Ä	Ä
S	S	S	S	T	H	I	I	I	I	I	E	E	E	Ä	Ä
S	S	S	S	H	I	I	I	I	I	I	I	E	E	E	Ä

Figure 2: *The Gaussian mean vectors of a 192-mixture example are organized into a 16x12 table by SOM, labeled by majority voting and trained futher by LVQ. The upper picture shows the cepstrums and the lower shows the corresponding phoneme labels. The $<$ and $>$ labels are for silence.*

more efficient models with fewer parameters than by for example continuous density models [8]. Diagonal covariance matrices are applied so that large mixtures, for example $K = 21$ and $M = 192$, can be used without having to estimate an excessive number of covariance parameters. The number of covariance parameters can be even further reduced by using tied covariance matrices between all Gaussians [8].

3 INITIALIZATION

The set of Gaussians which is used in mixture density functions of the states of the SCHMMs must be carefully chosen in order to create a suitable representation of the acoustical features produced by every phoneme. Since the estimation of the mixture weights and the state transition probabilities depends on the selection of the Gaussians the initialization of especially the mean vectors seems to be important when using large mixtures.

SOM [5] provides an efficient way to produce suitable values for the mean vectors. Compared to the more traditional K-means clustering the SOM seems to produce a more proper set of vectors for further training due to the use of neighborhoods in training which organizes the similar vectors close to each other and provides a kind of smoothing for the Gaussians.

In [7] it was observed that the final recognition error rate for CDHMMs was slightly decreased when the Gaussian mean vectors were modified by discriminative LVQ training. When applying the LVQ training for the SCHMMs the Gaussian densities must be first labeled to be able to determine the classes of the mean vectors. The labeling is done by majority voting where each training vector of each training sample gives one vote for its phoneme label to the nearest mean vector. The result of the illustrated 192-mixture model is shown in figure 2.

The experimental average error rates when applying Different LVQ algorithms [5] are compared in table 4.

4 TRAINING METHODS

The covariance matrices, mixture weights and transition probabilities can be effectively estimated by using the maximum likelihood methods. The basic reestimation algorithm is the iterative Baum-Welch [3] or the forward-backward algorithm. Another algorithm which aims at the same objective, i. e. to maximize the likelihood of the training data, is the segmental K-means algorithm (also called Viterbi training) [10].

When the Gaussian mean vectors are initially estimated by SOM the mixture coefficients of a state can be visualized by a three dimensional surface like in figure 3. It is easy to observe that the largest coefficients are grouped to the same areas as the labels in figure 2, as expected.

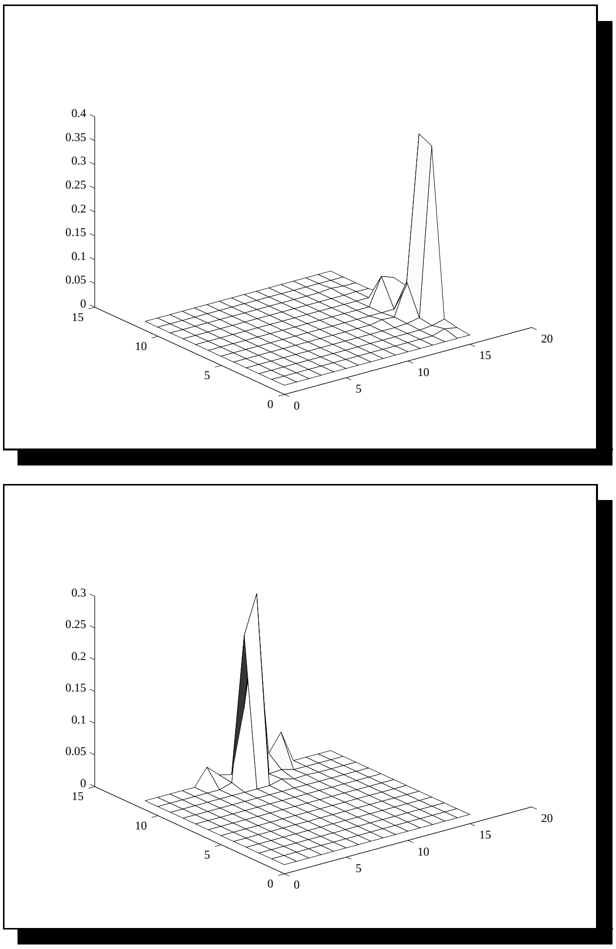


Figure 3: *The mixture coefficients in output density functions for states A2 and I1. The 16x12 codebook of Gaussians is the same as in figure 2.*

5 RECOGNITION AND ERROR CORRECTIVE TUNING

The recognition is performed by the Viterbi decoding, which gives the most probable path through the states of the SCHMMs that could have produced the acoustical short-time features computed from the spoken word. The error corrective tuning is based on the comparison of that path and the correct path (table 1). The correct path is defined as the most probable path that goes through the correct phonemes in the correct order. The LVQ2 learning law is used to move slightly the closest Gaussian mean vectors having the correct and the incorrect label where the most probable path differs from the correct path. The objective is that when a recognition error occurs the output density function of the correct state is tuned to represent better the current feature vectors and the density of the incorrect state is tuned to the opposite direction. As can be observed from table 3 this procedure decreases significantly the average error rate also for the test data.

```
LLLLLLAAAAAAAAAAEEEEEEEEEENNNEEEEEEEEEEEEENNNNNNNNNN
LLLLLLAAAAAAAAAAIIIIIIIIIINNNEEEEEEEEEEEEENNNNNNNNNN
L.......A........I.......N.....E..........N.........
```

Table 1: *The first row contains a part of the labels of states visited by the most probable path through a training sample of the word "OULAINEN". The second row is the corresponding most probable correct path obtained by the constrained Viterbi search. The last row presents the manually segmented starting points of the phonemes.*

6 EXPERIMENTS

The speech recognition system of the Laboratory of Information and Computer Science at Helsinki University of Technology [11] was used in the experiments. The system works like a phonetic typewriter transcribing spoken Finnish utterances into phoneme sequences. The acoustical features used as observations of speech signal are 20-component cepstral vectors computed every 10 ms from a 20 ms window concatenated with the energy of the signal. The experiments were carried out using a set of 311 Finnish words and three native Finnish speaker. From each speaker four repetitions of the word set were available and the speaker dependent SCHMMs were trained with three sets per speaker leaving one set for testing.

The recognition performance was determined by calculating the sum of changed, missing and extra phonemes in decoded phonetic transcriptions divided by the correct number of phonemes in the test material. To get the average recognition error rates presented in tables (2, 3 and 4) the four different training and test set combinations were applied for each speaker yielding a total number of 12 tests.

The statistical significance of the difference in error rates between different algorithms is verified by the Matched-Pairs test [1]. Since the number of independently pronounced segments (words) is large enough (3732) the difference in the number of errors in a segment made by the compared algorithms can be approximated by a normal distribution and the test is applicable.

The default parameters used in the presented experiments are:

- Configuration: 4-state SCHMMs with 192 mixtures
- SOM initialization: first 1 cycle with all training samples decreasing the neighborhood radius down from 10 (in a 16x12 array) and the learning rate down from 0.2 and then 5 cycles with radius down from 3 and learning rate down from 0.02
- LVQ initialization: 2 cycles with all training samples decreasing the learning rate down from 0.02 and using learning law LVQ3 with window 0.3 and $\epsilon = 0.1$
- Reestimation: 10 iterations of Baum-Welch algorithm

- Tuning: 1 cycle with all training samples decreasing the learning rate down from 0.02 and using learning law LVQ2 with window 0.3 applied when the most probable path differs from the correct path

states	mixtures		
	192	494	999
3	12.04	10.68	
4	10.02	8.63	7.93
5	9.45	8.27	

Table 2: *Recognition error rates for different SCHMM configurations.*

algorithm	states	error1%	error2%	risk
Baum-Welch	4	9.82	9.44	0.0005
Baum-Welch	5	9.14	8.99	0.15
Viterbi	4	12.17	11.43	0.0001
Viterbi	5	12.47	11.89	0.0001

Table 3: *Recognition error rates for different SCHMM training algorithms (error1%) and after the LVQ-based error correction (error2%). The risk level is from Matched-Pairs test and determines the risk by which the error corrective tuning decreases the error rate for that row.*

version	window	error%	risk
SOM		10.52	
SOM+LVQ1		10.34	0.22
SOM+LVQ2	0.3	9.95	0.0001
SOM+LVQ2	0.5	10.03	0.0001
SOM+LVQ3	0.3	10.02	0.0001
SOM+LVQ3	0.5	10.20	0.0096
*SOM+LVQ3	0.3	9.52	0.0001
KM		10.54	-0.89
KM+LVQ3	0.3	10.75	-0.17

Table 4: *Recognition error rates for variations of LVQ used in the initialization of the Gaussian mean vectors for 4-state SCHMMs. No tuning is performed after the Baum-Welch reestimation. In * the neighborhood radius is decreased only to 1. KM is the Kmeans algorithm. The risk level by the Matched-Pairs test on each row determines the risk by which that initialization method is statistically better than the method on the first row. The minus sign means that the method is worse by that risk level.*

7 CONCLUSIONS

The best results in the experiments were obtained by using SCHMM configuration of 4 states per phoneme and a mixture of totally 999 Gaussians (SOM size 37x27). However, the results obtained with 5 states per phoneme were generally better than with 4 states, which would encourage to test large mixtures also for 5-state models although the number of density parameters would be large. According to the experiments the Baum-Welch reestimation is more suitable for the SCHMMs than the Viterbi algorithm (Segmental K-means). The LVQ-based tuning decreases the recognition error rate significantly for some configurations, but some more careful investigations are still necessary for the development of that method for error corrective tuning. For the different variations of LVQ used for the discriminative initialization of the Gaussian mean vectors the best results were obtained with LVQ2 and the 0.3 window. Decreasing the neighborhood radius only to 1 in the SOM gave better starting values than decreasing to 0 or using Kmeans.

References

[1] L. Gillick and S.J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of ICASSP*, pages 532–535, Glasgow, Scotland, May 23-26 1989.

[2] X.D. Huang and M.A. Jack. Unified techniques for vector quantization and hidden Markov modeling using semi-continuous models. 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] Shigeru Katagiri, Chin-Hui Lee, and Biing-Hwang Juang. New discriminative training algorithms based on the generalized probabilistic descent method. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 298–308, Princeton, New Jersey, USA, October 1991.

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

[6] Takashi Komori and Shigeru Katagiri. GPD training of dynamic programming-based speech recognizers. *J.Acoust.Soc.Jpn*, 13(6):341–349, 1992.

[7] Mikko Kurimo and Kari Torkkola. Combining LVQ with continuous density hidden Markov models in speech recognition. In *Proceedings of the SPIE's Conference on Neural and Stochastic Methods in Image and Signal Processing*, volume 1766, pages 726–734, San Diego, USA, July 1992.

[8] 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.

[9] Shinobu Mizuta and Kunio Nakajima. An optimal discriminative training method for continuous mixture density HMMs. In *Proceedings of ICSLP*, volume 1, pages 245–248, Kobe,Japan, November 1990.

[10] L.R. Rabiner, J.G. Wilpon, and B.H. Juang. A segmental K-means training procedure for connected word recognition. *AT&T Technical Journal*, 64:21–40, 1986.

[11] 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.