CORRECTIVE TUNING BY APPLYING LVQ FOR CONTINUOUS DENSITY AND SEMI-CONTINUOUS MARKOV MODELS

Mikko Kurimo

Laboratory of Information and Computer Science
Helsinki University of Technology

Abstract

In this work the objective is to increase the accuracy of speaker dependent phonetic transcription of spoken utterances using continuous density and semi-continuous HMMs. Experiments with LVQ based corrective tuning indicate that the average recognition error rate can be made to decrease about 5% – 10%. Experiments are also made to increase the efficiency of the Viterbi decoding by a discriminative approximation of the output probabilities of the states in the Markov models. Using only a few nearest components of the mixture density functions instead of every component decreases both the recognition error rate (5% – 10% for CDHMMs) and the execution time (about 50% for SCHMMs). The lowest average error rates achieved were about 5.6%.

1 INTRODUCTION

Several suggestions have been recently published, describing training methods for HMMs using the minimization of the number of misclassifications directly as a training criterium. A formal way to realize this criterium is to define a continuous cost function that behaves like discrete number of misclassifications but can be differentiated and applied for stochastic gradient descent algorithm, e. g. the Generalized Probabilistic Descent (GPD) [3]. A more experimental way is to update the model parameters according to some heuristically motivated rules so that the correct classification would become more and the incorrect less likely [1, 8].

As proved in [5] the misclassification measure of GPD can be heavily simplified and updation rules resembling the Learning Vector Quantization (LVQ) [4] are obtained. In this paper several different versions of LVQ based corrective tuning methods are presented and tested by phoneme recognition experiments. The training method combinations presented for CDHMMs [7] and SCHMMs [6] are used in this work to create good initial models to act as a starting point for iterations of corrective tuning.

2 CDHMM AND SCHMM

The hidden Markov models describe each phoneme as a sequence of different states producing observations represented by a set of quantized output symbols or by a weighted mixture of parametric probability density functions. Both continuous density and semi-continuous HMMs apply the latter modeling approach. The main functional difference between them is that in the SCHMMs the mixture densities are tied between the states , i. e. each state has the same set of mixture densities. This tying requires a much larger set of mixture components, but since the components are common between the states, the total number of parameters to be estimated is smaller. In both HMMs tying the covariance matrices across all models allows more efficient use of the parameters because more mixture components can be estimated without increasing the number of parameters excessively. For CDHMMs 4 – 35 Gaussian mixtures per state are used and for SCHMMs the number of the tied mixtures is varied between about 200 and 1000.

3 TRAINING METHODS

The mean vectors of the Gaussian densities are initialized by clustering the feature vectors of the training samples by using the Self-Organizing Map [4]. The mean vectors can be further modified by LVQ to get more discrimination between the phonemes. In CDHMMs the state for which the Gaussian mean vector belongs corresponds the vector class in LVQ [7]. In SCHMMs the definition of the vector class is not so obvious because mixtures are common for all states. That problem is solved in [6] by labeling each Gaussian mean vector with a phoneme label by majority voting (Figure 1), where each training feature vector has one vote. This labeling method is similar as used with the discrete HMMs [9].

After the initial Gaussian mean vectors are chosen the HMMs can be completed by estimating the covariance matrices, mixture weights and transition probabilities and updating the mean vectors by maximum likelihood methods such as the iterative Baum-Welch reestimation and the Viterbi training (also called the segmental K-means algorithm).

<	<	<	<	<	<	<	<	>	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 1: The labels determined by majority voting for the Gaussian mean vectors used in SCHMMs. The $<$ and $>$ labels are for silences. Vectors are initially clustered by Self-Organizing Map (size 16x12 units).

4 CORRECTIVE TUNING

Error correction is intuitively an obvious way to enhance the recognition ability of the models since the objective is to reduce the number of recognition errors. Since a modification of the model to recognize one sample correctly may cause a misrecognition of another sample the models are tuned in small steps to reach a global minimum of errors. In corrective tuning the training samples are first recognized and when an error is encountered the model is slightly modified in order to achieve the correct transcription. Instead of searching for the global minimum of errors the training data is circulated only a couple of times using small modification steps to avoid an excessive adaptation to the training data.

The corrective tuning method presented here is based on the comparison between the most probable path through the states of the HMMs for a training sample and the most probable correct path. The most probable correct path is obtained by constraining the Viterbi search so that the correct phonemes are produced in the right order as on the second row of the Table 1. The manually marked starting points (3rd row) of phonemes can be also used to construct the correct path. For the time instants where the recognized path differs from the correct path the output density function of the correct state is altered to give a larger probability for the current feature vector. The output density function of the incorrect state can be respectively altered to the opposite direction.

In this work an output density function is modified by finding the closest Gaussian mean vector in the mixture density function and moving it in respect of the current feature vector according to the LVQ learning laws [4].

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

```
AAAAAAAAEEEEEEEEEENNNNEEEEEEEEEEEEENNNNNNNNNN
AAAAAAAAIIIIIIIIINNNNEEEEEEEEEEEEENNNNNNNNNN
A........I.......N.....E..........N.........
```

Table 2: Recognition error percentages for variations of corrective tuning. First row is errors before correction. The error rate is the sum of changed, missing and extra phonemes in decoded phonetic transcriptions divided by the correct number of phonemes for the test set. The parameters N and n are explained in the section 5.

Version	CDHMM, $N = 24$		SCHMM, $N = 192$	
	$n = 24$	$n = 1$	$n = 192$	$n = 10$
	6.8	5.9	9.6	9.5
v1	6.8	5.9	9.7	9.7
v2	6.8	5.6	9.6	9.5
v3	6.8	5.9	10.8	10.9
v4	6.4	5.5	9.8	9.9
v5	6.3	5.6	9.2	9.5

The experimentally tested variations (Table 2) of the method are:

- v1: Only the nearest mean vector is updated by LVQ1 learning law
- v2: The two nearest mean vectors are updated by LVQ2 learning law.
- v3: The two nearest mean vectors are updated by LVQ3 learning law.
- v4: For CDHMMs the nearest mean vector of the current state in the most probable path and in the correct path are updated by LVQ2 learning law. For SCHMMs the nearest mean vector with the current label in the most probable path and in the correct path are updated by LVQ2 learning law. Here, manual labeling results (3rd row in Table 1) are exploited to determine the correct labels.
- v5: As v4, but no manual labeling is used. The correct path is computed by using the constrained Viterbi search using only the information about the correct phonetic transcription of the training word.

An attempt to illustrate the corrective tuning method matching to the two latter versions (v4 and v5) is presented in Figure 2.

In addition to being intuitively appealing the corrective tuning by LVQ has also a strong theoretical

	Word	Labels on the best path
Incorrect	KEMU	KKKKEEEEEEMMMMMMMUUUU $\boxed{\text{U}}$ UUU
Correct	KEMI	KKKKEEEEEEMMMMMMMIIII $\boxed{\text{I}}$ III

Nearest incorrect (U) Gaussian is moved away and correct (I) closer

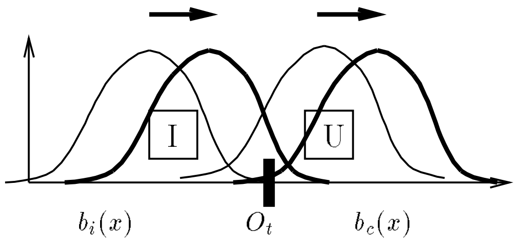

Figure 2: Corrective tuning method. The incorrect but most probable HMM path and the correct path through a training word are compared. For labels different in those two paths the output densities ($b_i(x)$ and $b_c(x)$) of corresponding states are modified by moving the nearest Gaussian mean vectors to the current feature vector.

background since it can be derived by simplifying the Generalized Probabilistic Descent (GPD) method [5] which in turn minimizes the continuous cost function generated by the misclassifications.

5 DENSITY APPROXIMATION IN RECOGNITION

The construction of the phonetic transcription of a spoken utterance by using continuous density HMMs consists of the following main phases:

1. The computation of the short time cepstral feature vectors (every 10 ms).
2. The computation of the observation probablilities for each feature vector for each Gaussian pdf used in the mixture density functions.
3. The formation of the output probabilities for each state by multiplying the Gaussian probabilities by the set of mixture coefficients.
4. The search for the most probable path through the states of HMMs using the Viterbi search to combine the output probabilities and the state transition probabilities.
5. If no transformations or postprocessing rules are used, the last task is the collection of the phonetic labels corresponding the Markov models along the path.

The output probability calculations (phase 2), which are made for each observed short time feature vector and for each state, consume a significant part of the total computation time. One way to reduce the computational load of these calculations is to approximate the output probabilities by using only the nearest Gaussians (using the distances between mean vectors and the current feature vector) instead of the whole set [2]. This is feasible because finding the set of n nearest vectors is a considerably lighter task than calculation of the full probabilities for the rest of the $N-n$ Gaussians if n is, for example, smaller than half of N where N is the total number of Gaussians used for the mixture. This approximation also tends to increase the discrimination between the states because only a couple of nearest Gaussians affects the output probability. The usage of only a few Gaussians emphasizes perhaps more clearly the distinctive features since when all the Gaussians are involved the robability reflects more about how well the current feature vector represents the average features of that state.

The Tables 3 and 4 show the error rates for different values of n and N in CDHMMs and SCHMMs, respectively.

Table 3: Recognition error percentages (e%) for various n and N values for CDHMMs. The bold-faced percentages indicate the best n values for each mixture size.

$N=35$		$N=24$		$N=15$		$N=9$		$N=4$	
n	e%	n	e%	n	e%	n	e%	n	e%
35	6.1	24	6.2	15	6.3	9	6.9	4	8.0
10	5.9								
7	5.9	7	6.1	5	6.1				
5	5.8	5	5.9	4	6.0				
3	**5.6**	3	5.8	3	6.0	3	6.7		
2	5.7	**2**	**5.8**	**2**	**5.9**	2	6.6	**2**	**7.9**
1	5.8	1	5.8	1	6.0	**1**	**6.6**	1	8.0

Table 4: Recognition error percentages (e%) for various n and N values for SCHMMs.

$N=999$		$N=494$		$N=352$		$N=192$	
n	e%	n	e%	n	e%	n	e%
999	**7.4**	494	8.2	352	8.6	192	9.6
200	7.4	**100**	**8.1**	**100**	**8.6**	80	9.6
100	7.5	50	8.2	50	8.6	40	9.5
50	7.6	25	8.3	25	8.7	**20**	**9.5**
25	8.1	15	8.8	18	8.8	15	9.5
15	9.3	10	9.5	10	9.4	10	9.5
10	12.2	5	14.7	5	12.4	5	10.6

6 EXPERIMENTS

The experiments were accomplished by the speech recognition system of the Laboratory of Information and Computer Science at Helsinki University of Technology [9]. 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.

7 CONCLUSIONS

For the different variations of LVQ used for corrective tuning the best were in these experiments LVQ2 with the correct path defined as the most probable path visiting the correct phonemes in the correct order. According to the experiments the discriminative probability approximation by using only n nearest Gaussians instead of the whole mixture decreases the recognition error rate for the CDHMMs compared to calculating observation probabilities using all N Gaussians of the mixture density functions. For the SCHMMs the suitable value seems to be $n = N/20$ which reduces the computational load significantly without increasing the error rate essentially.

References

[1] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. A new algorithm for the estimation of hidden Markov model parameters. In *Proceedings of ICASSP*, pages 493–496, 1988.

[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] 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.

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.