SEGMENTAL LVQ3 TRAINING FOR PHONEME-WISE TIED MIXTURE DENSITY HMMS

Mikko Kurimo

Helsinki University of Technology, Neural Networks Research Centre
Rakentajanaukio 2 C, FIN-02150, Espoo, FINLAND
tel: +358 9 451 3266, fax: +358 9 451 3277, email: mikko.kurimo@hut.fi

ABSTRACT

This work presents training methods and recognition experiments for phoneme-wise tied mixture densities in hidden Markov models (HMM). The system trains speaker dependent, but vocabulary independent, phoneme models for the recognition of Finnish words. The Learning Vector Quantization (LVQ) methods are applied to increase the discrimination between the phoneme models. A segmental LVQ3 training is proposed to substitute the LVQ2 based corrective tuning as a parameter estimation method. The experiments indicate that the new method can provide the corresponding recognition accuracy, but with less training and more robustness over the initial models. Experiments to upscale the current system by introducing context vectors and larger mixture pools show up to 40 % reduction of recognition errors compared to the earlier results in [10].

1 INTRODUCTION

The Learning Vector Quantization (LVQ) has been applied to the minimization of classification errors in the phoneme recognition right from its introduction in [4]. The hybrid discrete HMM–LVQ systems [2] use LVQ to train a vector quantization (VQ) codebook to transform the acoustic features into short-time phoneme symbols called "quasiphonemes". For continuous density HMMs with mixture Gaussian densities the LVQ was first used to improve the mixture initialization [11] and to enhance the discrimination between the maximum likelihood models by corrective training [12, 8]. It can be also shown that the LVQ2 learning law can be derived from a simplification of the Generalized Probabilistic Descent (GPD) training [3] to minimize classification errors on the training data [8], which relates the LVQ2 based corrective tuning [9] and the segmental GPD [15] under the same framework.

The idea of segmental HMM training (the same as Viterbi training) was developed [14] to provide an adequate and gradually improving segmentation of the training samples in order to determine the proper set of training features for each HMM state. In contrary to the conventional supervised neural network learning the segmental methods should tolerate an uncertain feature vector labeling in the beginning of the learning.

The phoneme-wise tied density codebooks can be seen as an advanced version of the semi-continuous HMMs [1] to save resources by not estimating a totally different set of density mixtures for each state. However, it is also an extension of the VQ for selecting multiple winners from each codebook weighted by their quantization errors [1]. The density codebooks are initialized by training small Self-Organizing Maps (SOMs) [6], producing a representative set of reference vectors for each phoneme. The smoothed mapping provided by SOM is a suitable initialization for the mixture density estimation [10].

The experiments on this paper consist of comparing the segmental LVQ3 training to the traditional Viterbi training with the corrective tuning [9] on a speaker dependent and vocabulary independent Finnish phoneme recognition system. The robustness of the training method is tested by varying the input features and adding more parameters to the real-time baseline system [10] to improve the modelling accuracy.

2 SEGMENTAL LVQ3

2.1 The basic learning laws of LVQ2 and LVQ3

For a randomly selected labeled training vector $\boldsymbol{x}$, the *two* best-matching codebook vectors m_c, $m_{c'}$ representing different classes are adjusted, if $\boldsymbol{x}$ appears to be near the border between the two classes. The latest version LVQ2.1 [5] requires that one of these two best-matching codebook vectors belongs to the class of the training vector and the other to an incorrect class.

Under these conditions the tuning occurs as follows:

$$\begin{aligned} \boldsymbol{m}_c(t+1) &= \boldsymbol{m}_c(t) + \alpha(t)[\boldsymbol{x}(t) - \boldsymbol{m}_c(t)] \\ \boldsymbol{m}_{c'}(t+1) &= \boldsymbol{m}_{c'}(t) - \alpha(t)[\boldsymbol{x}(t) - \boldsymbol{m}_{c'}(t)] \ , \end{aligned} \quad (1)$$

where $\boldsymbol{m}_c$ is from the correct and $\boldsymbol{m}_{c'}$ from the incorrect class. The teaching gain $\alpha(t) \in (0,1)$ is decreased monotonically.

The confirmation that $\boldsymbol{x}$ is near the border between these two classes is inspected by defining a window of relative width $w \in (0,1)$ so that the ratio of the distances $d_c = ||\boldsymbol{x} - \boldsymbol{m}_c||$ and $d_{c'} = ||\boldsymbol{x} - \boldsymbol{m}_{c'}||$ must fulfill

$$\frac{1+w}{1-w} > \frac{d_c}{d_{c'}} > \frac{1-w}{1+w} \ . \quad (2)$$

In LVQ3 [6] the learning is extended for cases where all $\boldsymbol{x}, \boldsymbol{m}_c$ and $\boldsymbol{m}_{c'}$ belong to the same class to enhance the long run behavior of the training process. For such cases the update rule is

$$\boldsymbol{m}_i(t+1) = \boldsymbol{m}_i(t) + \epsilon\alpha(t)[\boldsymbol{x}(t) - \boldsymbol{m}_i(t)] \;, \qquad (3)$$

where $i \in \{c, c'\}$ and $\epsilon \in (0,1)$ a stabilizing constant factor [7]. The value of ϵ should reflect the width of the window w in (2) so that with a narrow window the stabilizing learning steps (3) are small (i. e. ϵ is small).

2.2 LVQ in segmental training

The segmental LVQ3 training is, practically, a modification of the traditional Viterbi training (also called segmental K-means [14]). The iteration starts by a Viterbi search to find the most probable segmentation of each training word producing its phonetic transcription. Then the output probability density of each state along the path is updated to increase the likelihood of the state much like in a K-means epoch for fixed-label training. The suggested modification is to check also the differences to the segmentation obtained by assuming the word unknown. If differences exists, the mixture densities of the differing states are tuned to increase the likelihood of the correct state and decrease that of the incorrect state.

The output density of a state is tuned by finding the best-matching Gaussian mixture in the density codebook and changing its parameters (mixture mean, weight factor) to provide a better (or worse, respectively) fit. As explained in [10] the covariance matrices are tied over all HMMs and not altered in the iteration. Figure 1 presents a rough one-dimensional sketch of the change in the output densities due to the tuning process.

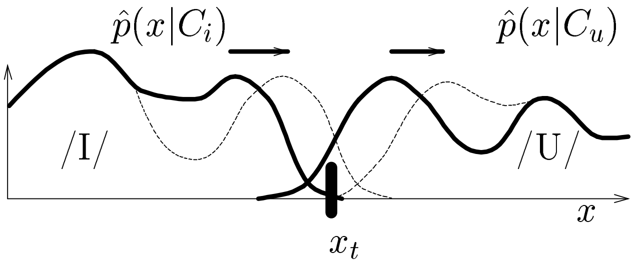

Figure 1: *Density tuning by applying LVQ. An approximative result on the mixture Gaussian output density functions $\hat{p}(x|C_u)$ and $\hat{p}(x|C_i)$ when the nearest Gaussian for the incorrect phoneme C_u is moved away and the nearest Gaussian for the correct phoneme C_i closer to the observation x_t.*

Due to the nature of the segmental training the output density modifications are effectively made in batch, i. e. the sets of updates collected in the batch variables are performed only after all words segmented by the current models are inspected. With the obtained new models the next improved segmentation can be sought.

The learning laws corresponding to LVQ3 (1,3) for the updates of the batch variables of the mixture means are obtained by adding the training vector $\boldsymbol{O}_t$ for the nearest mixture of the state on the correct path q (i.e. $\eta_t(i,m) = 1$) and subtracting it from the nearest mixture of the state on the best, but incorrect path $\bar{q}$ (i.e. $\nu_t(i,m) = 1$), if the segmentation by the best path is incorrect. The subtraction is performed by accumulating the batch variable by the vector $2\boldsymbol{\mu}_{im} - \boldsymbol{O}_t$ that is, by looking from the mixture mean $\boldsymbol{\mu}_{im}$, the one at the same distance as $\boldsymbol{O}_t$, but to the opposite direction. The indicator functions η and ν are defined as follows:

$$\eta_t(i,m) = \begin{cases} 1, \text{ if } q_t = i \text{ and } \boldsymbol{\mu}_{im} = \boldsymbol{\mu}_{io} \\ 0, \text{ otherwise} \end{cases} \qquad (4)$$

$$\nu_t(i,m) = \begin{cases} 1, \text{ if } \bar{q}_t = i \text{ and } q_t \neq i \text{ and } \boldsymbol{\mu}_{im} = \boldsymbol{\mu}_{io} \\ 0, \text{ otherwise}\;, \end{cases} \qquad (5)$$

where $\boldsymbol{\mu}_{io}$ is the mean vector of the closest Gaussian mixture of the state i to the observation vector $\boldsymbol{O}_t$. The indexes m and i run through all M mixture densities and for each state, respectively. One batch epoch consisting of the training sequences $l = 1, 2, \cdots, L$ (indexed by the superscript l) of variable sequence length T_l provides the new estimates of the mixture means:

$$\hat{\boldsymbol{\mu}}_{im} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}[\eta_t^l(i,m)\boldsymbol{O}_t^l + \nu_t^l(i,m)(2\boldsymbol{\mu}_{im} - \boldsymbol{O}_t^l)]}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}[\eta_t^l(i,m) + \nu_t^l(i,m)]} \;. \qquad (6)$$

Correspondingly, the batch variable of the mixture weight c_{im} is accumulated by the term $\eta(i,m) + \upsilon(i,m)\frac{c_{im}}{1-c_{io}}$ to tune the weight of the nearest mixture of the state on the correct path q (i.e. $\eta(i,m) = 1$) towards 1 and that of the state on the best, but incorrect path $\bar{q}$ (i.e. $\nu(i,m) = 1$) towards 0, if the segmentation by the best path is incorrect. To maintain the normalization, the weight of the nearest mean vector is not directly tuned towards 0, but instead the weights of the other Gaussians (ν is then replaced by υ) are tuned towards 1 by the fraction representing to their relative significance to the current state.

$$\upsilon_t(i,m) = \begin{cases} 1, \text{ if } \bar{q}_t = i \text{ and } q_t \neq i \text{ and } \boldsymbol{\mu}_{im} \neq \boldsymbol{\mu}_{io} \\ 0, \text{ otherwise}\,. \end{cases} \qquad (7)$$

The new estimates of the mixture weights will then be:

$$\hat{c}_{im} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}[\eta_t^l(i,m) + \upsilon_t^l(i,m)\frac{c_{im}}{1-c_{io}}]}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\sum_{m=1}^{M}[\eta_t^l(i,m) + \nu_t^l(i,m)]} \;. \qquad (8)$$

The composition of the proposed estimates differ slightly from the original definition of LVQ3 for the fixed-label training vectors (1–3), because the optimality of (1–3) was guaranteed for the simple nearest neighbor classification decisions. Anyhow, because of the tied covariances, the phoneme dependent density codebooks and the k-nearest mixture approximation [9], the applied form of mixture Gaussian densities is not so far from that simple decision making as one might first expect. Another difference from (1–3) is that the second-best mixture will not be modified in the correct classification case, but actually, that amendment is both simple and apparently harmless for the convergence. The convergence properties of the segmental LVQ3 approximate those of the LVQ learning in general [6] for fixed labels and since LVQ training is contained in the GPD framework as an extreme case by substituting L_∞ norm for the class distance [8], the convergence properties of the segmental minimum error classification training supports also the discrimination by segmental LVQ.

2.3 The differences between the segmental LVQ3 and the corrective tuning by LVQ2

The differences between the segmental LVQ3 and the earlier presented corrective tuning by LVQ2 [10] (also called segmental LVQ2) are, in addition to the basic distinctions in the LVQ3 and LVQ2 learning laws, the batch mode and the tuning of the state transition probabilities and the mixture weights. In the tuning based on LVQ2 the density codebook changes occur only due to misclassified phonemes, which makes it suitable mainly for fine tuning requiring traditional methods to provide the good initial models. The corrective tuning was applied in the stochastic mode controlled by a monotonically decreasing learning rate parameter, which also supports the fine tuning property, because the gradually improving segmentations complicate the determination of the learning rate decreasing schedule. The simultaneous adjustments of the mixture weights in the segmental LVQ3 training is yet another reason to eliminate the need for the separate initial training by the conventional maximum likelihood methods.

Although the LVQ2 based corrective tuning is a representative of the GPD discriminative training framework, it is a highly specialized one, and also other methods more much like the segmental LVQ3 could be derived in that framework. The difference is still, however, that no rival mixture densities representing false classifications for correctly recognized phonemes are tuned away from the area of observed features. This was prevented, because, if some mixtures get excessively pushed away during the first training epochs due to poor initialization or segmentation, the effects on the model convergence can be unfavorable. Another difference is that in many GPD applications for word or sentence recognition the magnitude of the parameter modification depends on the exact extent of whole word or sentence misclassification defined according to the log-likelihood scores of the state sequences [15].

3 EXPERIMENTS

The experiments involve testing the phoneme recognition accuracy for four sets of 350 Finnish words uttered by each of the three speakers chosen from the speech database of the Neural Networks Research Centre at the Helsinki University of Technology. By leaving one set out of four at a time for testing, an average error rate of totally 12 independent test runs is obtained. The average phoneme error rate is the sum of insertion, deletion and substitution errors divided by the correct number of phonemes in the test data.

The basic short-time acoustical feature vectors computed in every 8 ms from a 256 point FFT include 20 cepstral coefficients concatenated with the energy of the signal (the "normal feature vector" in Table 1). Context vectors [13] are formed by using concatenated averages of several successive short-time features (see Figure 2).

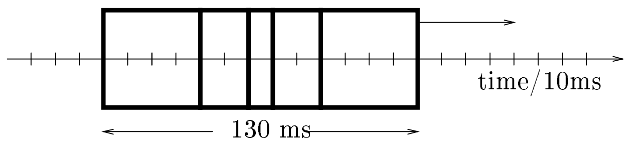

Figure 2: *The context vector includes concatenated averages of one, two and four successive cepstral vectors. Successive context vectors are formed by sliding the window above through the computed short-time features.*

To check the robustness of the method on different data and to be able to compare it with earlier results [10], the tests were also made for three speakers of an older database (called "Data 90" to distinguish it from the new "Data 95" in Table 1). The old data was recorded with a slightly lower sampling rate (12.8 vs. 16 kHz) and with shorter sessions (311 words), but the selected speakers were more experienced with the dictation, in average.

4 RESULTS

The main results refer to the statistical signification tests in Table 1. (a) Five training epochs is good enough in the segmental LVQ3, because, although the results still slightly improve between five and ten epochs, this difference is not statistically significant. The obtained results correspond then to those by the previously best system (the base line Viterbi training with additional corrective LVQ2 tuning [10]), but the required amount of segmentation-estimation epochs on the training data is only about one third of the previous system. (b) However, for the old system the recognition errors decrease from five to ten training epochs of traditional segmental K-means training and the decreasing continues, when the corrective tuning by LVQ2 is applied. (c) The comparison of the results of the segmental LVQ3 to the results of the segmental K-means reveals that when started from the same initial vales the LVQ3 converges considerable faster to the good error rates. After the first five training epochs the difference is already significant and, if the LVQ training is stopped, but the K-means still iterated for another five iterations, the results by the segmental K-means have still not yet improved enough to beat those of the segmental LVQ3.

Scaling up the system by increasing the dimension of the feature vectors and the number of mixture densities give about 40% lower error rates than the lowest error rates reported earlier [10] for the Data 90 (3.1% compared to 5.5%), but with the cost of increased recognition time per word by the factor around 6.

5 CONCLUSIONS

A novel training method for the mixture density HMMs for phoneme recognition is presented and compared to previous methods with experiments on two different Finnish databases. The segmental LVQ3 training links together the traditional Viterbi training by the segmental K-means algorithm and the corrective tuning by LVQ into an efficient training algorithm. Using the average results from speaker dependent tests on three differ-

Feature vectors	Mixt /phon	Error rate % on Data 95			
		skm	+slvq2	slvq3	+slvq3
normal	70	7.4	7.2	7.4	7.3
normal	140	7.2	7.1	7.1	6.9
context	70	4.9	4.6	4.9	4.8
context	140	-	-	4.5	4.3
Error rate % on Data 90					
normal	70	5.7	5.5	5.6	5.5
normal	140	5.3	5.4	5.3	5.2
context	70	3.6	3.4	3.5	3.4
context	140	-	-	3.3	3.1

Test	A	B	MP risk	MN risk
a	10x slvq3	slvq3	>0.05	>0.05
	skm + slvq2	slvq3	>0.05	>0.05
b	skm + slvq2	skm	0.005	0.005
	skm	5x skm	0.025	0.005
c	slvq3	5x skm	0.05	0.005
	slvq3	skm	>0.05	>0.05

Table 1: *Average phoneme recognition error rates for Viterbi training by segmental K-means, LVQ3 and LVQ2. The default amount of training epochs is 10 for SKM and 5 for SLVQ3 and SLVQ2. The average error rate by training A is lower than by B, but the difference is statistically significant only if the risk level is not larger than 0.05 in Matched-Pairs (MP) and McNemar's (MN) statistical significance tests. The risk levels are the average of the results on Data 95. Experiments "-" were not made. Labels "a,b,c" refer to the text.*

ent speakers and different output density codebooks for HMMs the conclusion is that the new method provides lower error rates than the conventional Viterbi training and similar rates as the additional LVQ2 based tuning, but with a considerably smaller training effort.

REFERENCES

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

[2] 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.

[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. Learning vector quantization for pattern recognition. Technical Report TKK-F-A601, Helsinki University of Technology, 1986.

[5] Teuvo Kohonen. Improved versions of Learning Vector Quantization. In *Proceedings of the International Joint Conference on Neural networks (IJCNN)*, volume 1, pages 545–550, San Diego, California, June 1990.

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

[7] Teuvo Kohonen. *Self-Organizing Maps.* Springer, Berlin, 1995.

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

[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. Hybrid training method for tied mixture density hidden Markov models using Learning Vector Quantization and Viterbi estimation. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 362–371, Ermioni, Greece, September 1994.

[11] 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.

[12] 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.

[13] Jyri Mäntysalo, Kari Torkkola, and Teuvo Kohonen. LVQ-based speech recognition with high-dimensional context vectors. In *Proceedings of the International Conference on Spoken Language Processing*, volume 1, pages 539–542, Banff, Canada, October 1992.

[14] 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.

[15] W.Chou, B.H. Juang, and C.H. Lee. Segmental GPD training of HMM based speech recognizer. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 473–476, Baltimore,USA, april 1992.