SPEECH RECOGNITION USING TEMPORALLY CONNECTED KERNELS IN MIXTURE DENSITY HIDDEN MARKOV MODELS

Panu Somervuo

Helsinki University of Technology
Neural Networks Research Centre
P.O. Box 5400 FIN-02015 HUT
Finland
panu.somervuo@hut.fi

ABSTRACT

A method is presented for speeding up the performance of the HMM based speech recognition system where the states are modeled by a large number of Gaussian kernels. The emission probabilities of the states are usually dominated by the nearest Gaussians to the input vector. The speedup is gained without deteriorating the recognition accuracy by concentrating on these kernels in the reduced K-best-kernel search. In this work, the time information of the input is encoded to the connections of the kernels. The search for the dominating kernels is then performed along the kernel connections which model the trajectories of the speech in the feature space. In the experiments, speaker-dependent speech recognizers were trained for ten speakers. The number of distance computations between feature vectors and kernel mean vectors was reduced 75% without increasing the average phoneme recognition error, which was 5.7% for the baseline system.

1. INTRODUCTION

Most speech recognizers today are based on hidden Markov models (HMMs) [6]. An HMM based approach gives, however, only a general framework and leaves many practical modeling problems open. When aiming at real-time speech recognition, the simplicity and speed of computation are the criteria which guide the design of the recognizer.

The speech recognition system used in this work is based on the HMMs using phoneme-wise Self-Organizing Maps (SOMs) [1, 2] as the basis of the probability density functions [4].

The SOM [1, 2] is an artificial neural network which defines a nonlinear transform from the input space to the set of nodes in the output space. Each node is associated with a model of the input space. Through an unsupervised learning process, the models become specially tuned and organized according to input patterns smoothly approximating the distribution of the input data. In its basic formulation, the SOM algorithm organizes static, separate feature vectors according to their similarity, and no temporal dependencies of the input items are taken into account.

Hidden Markov models (HMMs) are models of sequential data [6]. Their benefits in speech recognition are storing the temporal speech patterns compactly in a state network and utilizing the time-dependency and order of acoustic phenomena in recognition. The HMM is defined as a triple

$$\lambda = (A, B, \boldsymbol{\pi}), \tag{1}$$

where $A = [a_{ij}]$ is an $N \times N$ matrix of the state transition probabilities, $B = \{b_i\}_{i=1}^{N}$ is a set of emission probability density functions (pdfs) of N states, and $\boldsymbol{\pi}$ is an initial state probability vector.

This paper concentrates on modeling the trajectories of the speech inside HMM states and utilizing this for speeding up the recognition. The recognition system is based on modeling the pdfs by means of a large number of Gaussian kernels. The initialization of the pdfs is done by training a SOM for each phoneme [4]. Each model vector of the SOM becomes then a mean vector of the Gaussian mixture. The traditional Viterbi decoding is used in recognition for obtaining the best state sequence [6]. The speedup results from reducing the amount of computation when determining the emission probabilities of the states.

When modeling the pdf as a mixture of Gaussians, the emission probability for a given input feature vector is usually dominated by the few Gaussians only [4]. The amount of computation can thus be reduced by selecting only the subgroup of the Gaussians of the whole mixture and then computing the emission probability of the state by means of these kernels.

Previous work on speeding up the search for finding the dominating Gaussians on the SOM-codebook has been presented in [3]. In that work the reduced search was based

on the ordering of the feature vectors on the SOM without using any time information.

In the current work, the time information of the input is taken into account when forming the connections between the nodes of the SOM. The connection is created between those two nodes which are the best-matching units for two successive input items in time [8]. The node connections follow then the temporal trajectories of the speech in the feature space, see Fig.1. The reduced search for K nearest Gaussians is obtained by investigating only the nodes connected from the previous best-matching unit. Depending on the number of the connections, the savings can be considerable compared to the full codebook search where all kernels have to be investigated at every speech frame.

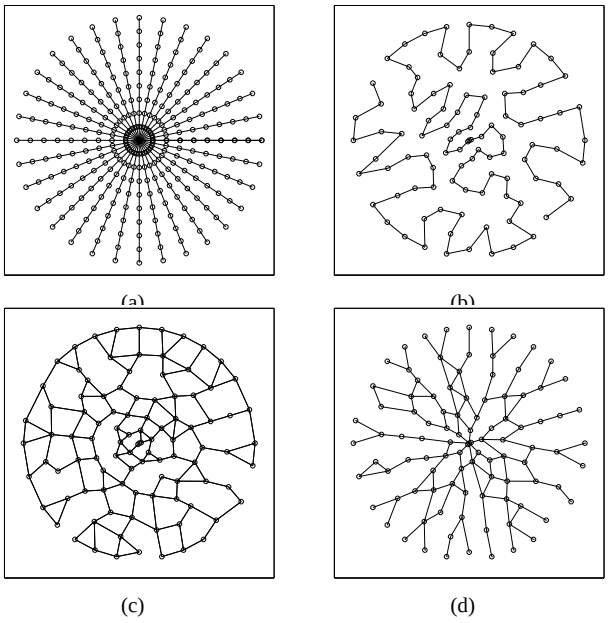

Figure 1: An illustrative experiment of different codebook vector connections. Input data consist of sequences of two-dimensional feature vectors proceeding from the origo to the unit circle (a). Input items are depicted by dots and successive input items in time are connected by lines. A one-dimensional SOM with 100 nodes was constructed using this data (b). The model vectors are depicted by dots and the neighborhood connections are depicted by line segments. Fig. (c) shows the connections created between two nearest nodes in the input space for each input item. Model vectors are the same as in (b). Fig. (d) represents the connections created between the best-matching units of two successive input items in time. The model vectors are the same as in (b) and (c). The original input data does not consist of separate feature vectors only, but sequences of them with time information. Network (d) resembles thus clearly best the original input data (a).

2. METHOD

The recognition system is based on phoneme-wise HMMs. Each phoneme is modeled by a three-state left-to-right HMM. A separate SOM is trained for each phoneme and a spherical Gaussian kernel is attached to each SOM node [4]. A fixed kernel width is used as a smoothing parameter of the final pdf. After initializing the state codebooks by the SOMs, the HMMs are trained by the segmental k-means algorithm [6]. Only state transition probabilities inside phoneme models are re-estimated. The probabilities for phoneme transitions have been estimated from the larger Finnish text corpus.

The use of temporal context of the short-time feature vectors has been found to improve the robustness of the recognizer and the discrimination of phoneme classes [5]. The feature vectors used in this work were concatenations of three sine-liftered 12-dimensional mel-cepstrum vectors concatenated at 50 ms time intervals [7]. These context vectors were computed every 10 ms.

Spherical Gaussians with fixed kernel widths have given good results in speech recognition earlier [4]. The limited amount of training data makes it difficult to get robust estimates for full covariance matrices, and furthermore, in speech recognition the error rate is more important than the likelihood of the model. The single variable for the width of the Gaussian is easier to tune than the full covariance matrix. The effect of the kernel width and the number of the Gaussians used in the computation of the emission probabilities of the states were experimented using 12-by-8-unit SOM codebooks for phonemes, see Fig. 2. It is interesting that the best results were obtained using as few as two or three dominating Gaussians of the whole mixture in each state for each input vector.

The time information of the speech is encoded to the connections of the kernels. The connection is created between those two kernels which are the best-matching units for two successive input vectors in time. The strength of the connection between kernels i and j is

$$a'_{ij} = \frac{\sum_{t=2}^{T} \delta(i - c(\boldsymbol{x}_{t-1}))\delta(j - c(\boldsymbol{x}_t))}{\sum_{t=2}^{T} \sum_{j'} \delta(i - c(\boldsymbol{x}_{t-1}))\delta(j' - c(\boldsymbol{x}_t))}, \qquad (2)$$

where

$$\delta(l) = \begin{cases} 1 & \text{if } l = 0, \\ 0 & \text{otherwise} \end{cases}, \qquad (3)$$

$\boldsymbol{x}_t$ is the input vector at time t, and $c(\boldsymbol{x}_{t-1})$ and $c(\boldsymbol{x}_t)$ are the best-matching units, i.e., the nearest kernels, for successive input items $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$, respectively:

$$c(\boldsymbol{x}_t) = \arg\min_k ||\boldsymbol{x}_t - \boldsymbol{m}_k||^2, \qquad (4)$$

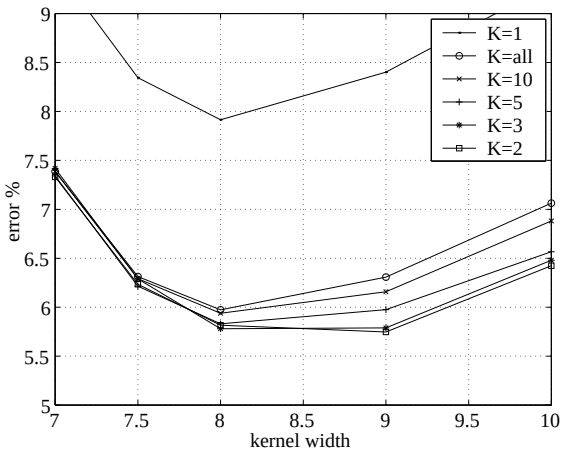

Figure 2: Average phoneme error rates of ten speaker-dependent speech recognizers with different values for the width of the Gaussians (horizontal axis) and the number of kernels K used in the state emission probability computation (separate error plots from bottom to top with K = 2,3,5,10,96, and 1). The number of Gaussians in each phoneme-wise codebook was 96. Three concatenated 12-dimensional cepstrum vectors were used as feature vectors.

where $\boldsymbol{m}_k$ denotes the centroid of the kth kernel, i.e., a model vector of the SOM.

Each node is provided by the list of nodes sorted according to equation (2). This is utilized in the reduced search of K-best Gaussians in the recognition phase. If node i has been the best-matching unit for input item $\boldsymbol{x}_{t-1}$, the K-best Gaussians for input item $\boldsymbol{x}_t$ can be searched using the whole list or only the first few elements in the sorted list of that node.

3. EXPERIMENTS

The speech data for phoneme recognition experiments was collected from six male speakers and four female speakers. Each speaker had uttered four times the vocabulary of 350 Finnish words. The speaker-dependent speech recognizers described in the previous section were trained using three speech sets. Each phoneme model consisted of three states which shared the kernels of the 12-by-8-unit SOM codebook with different weights. The phoneme-wise SOM codebooks were trained separately before the final HMM training which consisted of five cycles of segmental k-means algorithm. The phoneme recognition error was then computed using the fourth speech set. For the baseline system the average error was 5.7%.

In the following experiment, the effect of the interval of the full codebook search in the recognition was investigated. Two best Gaussians were used for computing the state emission probabilities in each state for each input vector. All parameters were kept the same as they were in the baseline system, only the connections between the kernels were added according to equation (2) using the training data. The connections were not restricted to be inside the phoneme-wise codebooks. It was found important to follow the speech trajectories along the whole training utterance and create the connections also for phoneme transitions from one phoneme-wise codebook to another. Otherwise the interval of the full codebook search could not have exceeded the duration of a phoneme without deteriorating the recognition accuracy. The recognition results for the test data are shown in Fig. 3. The number of corresponding distance computations are shown in Fig. 4.

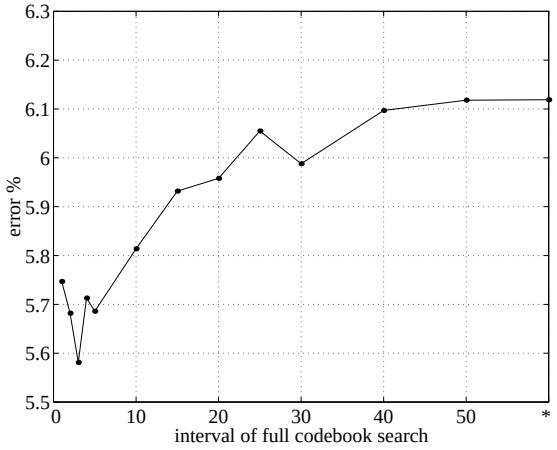

Figure 3: Average phoneme errors using the reduced two-best-kernel search for different intervals of the full codebook search. The rightmost error rate (*) is for the case where the full codebook search was performed only once in the beginning of each word utterance. The average length of a word utterance was 103 frames.

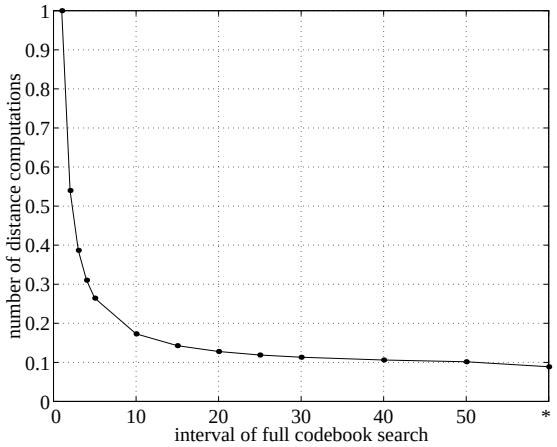

Figure 4: The number of distance computations between feature vectors and kernels. The values are scaled by the baseline system where the full search was performed at every speech frame (full codebook search interval 1).

On an average there were eight connections leaving from each node. This rather small number of connections can be explained by the use of context vectors as features. The connections became more specific than randomly scattered because the context vectors already carry information of the trajectory of the speech. However, if the number of the connections would become too large, the number of randomly scattered connections can be decreased by following only those connections whose strength computed according to equation (2) exceeds a given threshold.

75% reduction in the distance computations was achieved when the error rate remained below the error rate of the baseline system using the full codebook search at every speech frame. The lowest error rate, 5.6%, was achieved when the interval of the full codebook search was three frames.

For comparison, the speech recognition was experimented after reducing the size of the SOM from 96 units to 48 units. Full codebook search was performed at every speech frame. Using 8-by-6-unit SOMs, the average phoneme error was 7.8% the computational savings of distance computations being only 50% compared to the 96-unit SOMs. The recognition results were thus inferior compared to those of using the large SOMs but less distance computations. Because the larger SOM gave better results than the smaller SOM, the reason for the good recognition accuracy after reducing the number of distance computations was not that the large SOM contained unused codebook vectors which were constantly ignored in the reduced two-best-kernel search. The reason for the good performance was that the node connections modeled the temporal trajectories of the speech which were followed during recognition.

Kernel smoothing along the speech trajectories was also experimented, but this did not lead to any improvements in the recognition accuracy. Therefore, the best training procedure consisted of first initializing the phoneme-wise codebooks by the SOM, then training the HMMs by segmental k-means algorithm, and then creating the kernel connections.

4. SUMMARY

The emission probabilities of the HMM states for a given input vector are usually dominated by the nearest kernels of the whole mixture density. This gave the motivation to model speech trajectories inside the states and speed up the recognition phase of the already trained speech recognizer. Model vectors of the Self-Organizing Map were used as the kernel centroids of the mixture density and the speech trajectories were modeled by means of the kernel connections.

A connection was created between those two kernels which were the best-matching units for two successive input vectors in time. The reduced search for the two nearest Gaussians was obtained by investigating only the nodes which were connected from the previous best-matching unit in each phoneme-wise codebook.

In speech recognition experiments, the average phoneme recognition error did not increase from the baseline error, 5.7%, when the amount of distance computations between feature vectors and kernels was reduced 75%. In the baseline system the full codebook search was performed at every speech frame. Even when the reduction in the number of distance computations was over 90%, the error rate increased only to 6.1%. In that case the full codebook search was performed only in the beginning of each word utterance.

Modeling the trajectories of the speech by means of a large number of temporally connected kernels and following these trajectories during the recognition was demonstrated to be computationally efficient and to give very good results.

5. REFERENCES

[1] T. Kohonen. “Self-organized formation of topologically correct feature maps”. *Biological Cybernetics*, 43:59-69, 1982.

[2] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.

[3] M. Kurimo and P. Somervuo. “Using the Self-Organizing Map to Speed Up the Probability Density Estimation for Speech Recognition with Mixture Density HMMs”. *International Conference on Spoken Language Processing*, Philadelphia, PA, USA, pp. 358-361, 1996.

[4] M. Kurimo. “Using Self-Organizing Maps and Learning Vector Quantization for Mixture Density Hidden Markov Models”. *PhD thesis*, Helsinki University of Technology, Finland, 1997.

[5] J. Mäntysalo, K. Torkkola, and T. Kohonen. “Mapping context dependent acoustic information into context independent form by LVQ”. *Speech Communication* 14(2):119-130, 1994.

[6] L. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. *Proceedings of the IEEE*, 77(2):257-286, 1989.

[7] P. Somervuo. “Speech recognition using context vectors and multiple feature streams”. *Master’s Thesis*, Helsinki University of Technology, Finland, 1996.

[8] P. Somervuo. “Time Topology for the Self-Organizing Map”. *International Joint Conference on Neural Networks*. Washington DC, USA, July 1999.