*International Conference on Spoken Language Processing (ICSLP'92), Banff, Canada, October 12-16, 1992, Vol. 1, pp. 539-542*1

LVQ-based Speech Recognition with High-Dimensional Context Vectors

Jyri Mäntysalo, Kari Torkkola† and Teuvo Kohonen
Helsinki University of Technology, Laboratory of Information and Computer Science
Rakentajanaukio 2 C, SF-02150, Espoo, FINLAND
†IDIAP, CP 609, CH-1920 Martigny, SWITZERLAND, email: karit@idiap.ch

Abstract

In this paper we have applied the Learning Vector Quantization methods, including the latest developments [1, 3, 2, 4], to the task of Finnish speaker-dependent speech recognition. The main objective was to study the effect of radically increasing the dimensionality of the context vectors. The high-dimensional feature vectors in our work represent the whole phoneme and they are formed by both averaging and concatenating short-time feature vectors within a time domain window. Excellent results are achieved in separate phoneme classification of Finnish speech. Moreover, we also show how this method can be applied in combined labeling and segmentation of continuos speech. In this task we use an additional segmentation LVQ-codebook, and combine the information using HMMs.

1 Introduction

Theoretically, a single centralized classification decision is expected to be more accurate than any fusion of distributed local decisions due to information quantization effects. That is, in order to get best recognition performance, all the relevant information should be included into each recognition stage. With the task of separate phoneme classification using the Learning Vector Quantization methods [1, 3, 2, 4], the most natural way of applying this principle is to combine successive short-time feature vectors to form "context vectors". The high-dimensional feature vectors then represent the whole phoneme. In this work, they are formed by both averaging and concatenating short-time feature vectors within a time domain window. Since there exist no known theoretical way of deciding the combination that will give the most accurate classification results with the given data, we have tried to find the best form experimentally.

Because classifying pre-segmented phoneme locations is a problem of only an academic interest, we have also extended our phoneme center recognition experiments into combined labeling and segmentation of continuous speech. Our solution is to try to exploit vector quantization errors in addition to the phonemic classification results. Since the model vectors have been trained by samples from the phoneme centers, the locations of moving-average minima and maxima of quantization errors contain valuable information for segmentation. To the segmentation task, too, the LVQ has been applied whereby the context vectors are classified into two classes only, representing the stationary regions and transitions, respectively.

A wide choice of methods to exploit the information gathered by the LVQ methods is available, ranging from simple heuristic rules to mathematically justified Hidden Markov Models. We have concentrated on an approach, where above mentioned classification results are interpreted as output symbols of discrete-observation HMMs [6]. The models can also include quantization errors as continuos-density probability distributions. It should be emphasized that in this work the objective has not been to create a real-time speech recognition system, since the high-dimensional context vectors require much more computation time than e.g. the system described in [6].

2 Speech data and preprocessing

In the following experiments the same Finnish speech database has been used as in [6]. It contains four repetitions of a set of 311 words spoken by three male Finnish speakers. Each set contains 1737 phonemes. In the separate phoneme recognition experiments the unvoiced plosives /k,p,t/ have been divided into two parts, representing the silence part and the actual plosive, so that the real number of pre-segmented phonemes is now 1903. Three of the repetitions were used each time for training, and the remaining one for testing. Four independent runs were made by leaving one set at a time for testing, so that all the following recognition results are averages of 12 runs. As features computed every 10 ms we use 20-component mel-scale cepstra liftered with raised sine [6].

3 LVQ-based separate phoneme recognition

3.1 Application of the LVQ

We have first run an excessive amount of preliminary experiments to find the best way to apply the Learning Vector Quantization methods [1, 3, 2] to this specific task. We used the public domain software package *Lvq_Pak* [4]throughout these experiments. The following learning scheme proved to give the best results on average:

1. (Lvq_Pak: program 'eveninit') The codebook vectors have been initialized with samples from teaching data, so that each class contains an identical number of codebook vectors. All the candidates are tested against all the other samples; if the classification with the K-nearest-neighbor –method is correct, then the candidate will be accepted as a codebook vector initial value. This guarantees, that all the initial values are situated inside the correct class borders.
2. (Lvq_Pak: program 'balance') The medians of the shortest distances between the initial codebook vectors of each class are first computed. If the distances turn out to be very different for the different classes, new codebook vectors are added to or old ones deleted from the deviating classes, and a tentative training cycle based on the optimized-learning-rate LVQ1 algorithm is run once.

3. (Lvq_Pak: program 'olvq1') The codebook vectors are taught iteratively with the OLVQ1-algorithm using 10000 iterations and learning-rate initial values $\alpha_i(0) = 0.3$.
4. (Lvq_Pak: program 'lvq1') The codebook vectors are taught iteratively with the LVQ1-algorithm using learning-rate initial value $\alpha(0) = 0.2$ and decreasing it linearly during the learning process. Total number of iterations is 100 times the number of codebook vectors.

Four different codebook sizes, namely 200, 500, 1000 and 2000 units, were used.

3.2 Averaging experiments

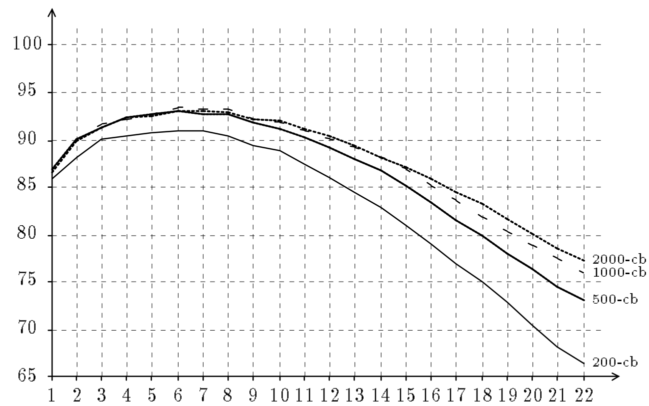

Figure 1: *Recognition accuracies as a function of the number of averaged short-time feature vectors.*

We first studied experimentally the effect of pure averaging the successive feature vectors starting from the pre-segmented phoneme centers. The recognition results are plotted in Fig. 1. The best results are obtained when the number of averaged feature vectors is approximately six. Thus the average width of the stable region can be approximated to be near 60 ms.

3.3 Concatenation experiments

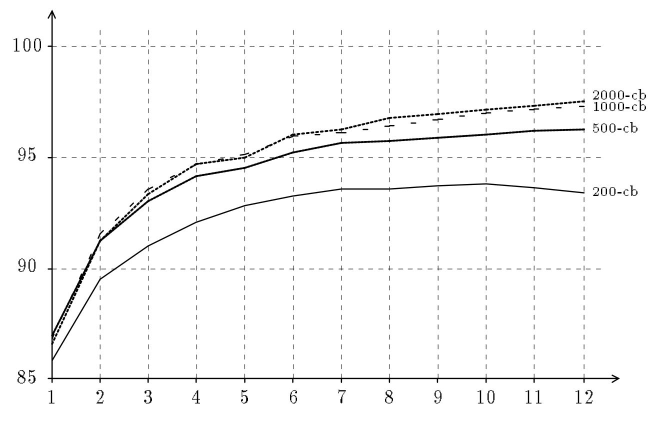

Figure 2: *Recognition accuracies as a function of the number of concatenated successive short-time feature vectors.*

Next, we have tested the effect of pure concatenation of successive short-time feature vectors into one context vector. Fig. 2 displays the results.

We have restricted our context vector dimensions to be less than or equal to 240 because of computation time limitations. To include more time context, we next try to concatenate every second short-time feature vector (Fig. 3).

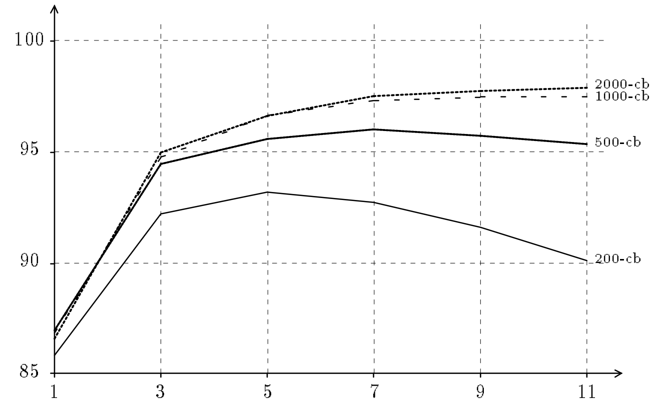

Figure 3: *Recognition accuracies as a function of the number of concatenated short-time feature vectors – every second vector is excluded.*

Excluding every second short-time feature vector does not significantly decrease the accuracies. Even better results can be expected if we concatenate averages of two or more successive feature vectors.

3.4 Experiments with combined concatenation and averaging

Next we have tested the effect of combined concatenation and averaging of successive short-time feature vectors into one context vector. The first graph (Fig. 4) shows the concatenation results, when the number of averaged feature vectors is two.

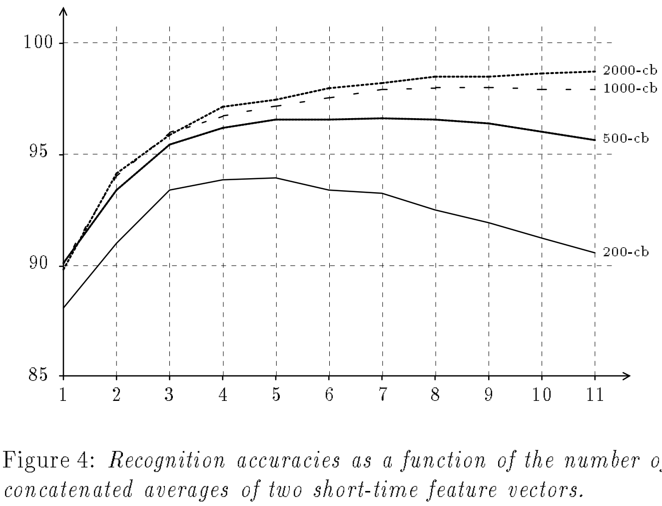

Figure 4: *Recognition accuracies as a function of the number of concatenated averages of two short-time feature vectors.*

In the next graph (Fig. 5) we have used averages of three feature vectors.

It is obvious, that outside the stable regions of the phonemes, speech signal contains more irrelevant information than in the phoneme centers due to coarticulation effects. Also, different durations of phonemes decrease the possibilities to exploit this information effectively. Thus it seems reasonable to use wider averages outside the stable regions. Table 1 lists the numerical

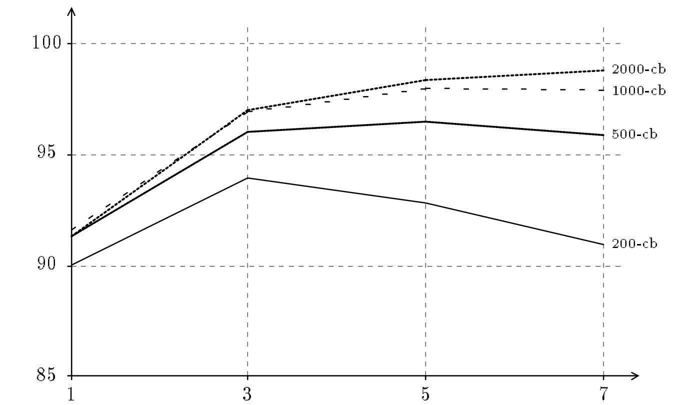

Figure 5: *Recognition accuracies as a function of the number of concatenated averages of three short-time feature vectors.*

values of the new experiments, where averages of four have been tried. For comparison, some previous results are included.

Context vector	cb:500 correct%	cb:1000 correct%	cb:2000 correct%
I	86.96	86.87	86.61
[....]	93.01	93.43	93.12
IIIIII	95.23	95.97	96.04
[] [] []	95.49	95.96	95.93
[..] [] [] [] [..]	97.03	97.93	98.22
[..] [..] [] [] [] [..] [..]	97.13	98.50	98.95
[] [] [] [] [] [] [] [] [] [] []	95.71	97.91	98.76
[.] [.] [.] [.] [.] [.] [.]	95.89	97.90	98.81

Table 1: *Numerical separate phoneme recognition results.* I *is a single short-time feature vector,* [] *means average of two,* [.] *average of three,* [..] *average of four, and so on. For example,* [..][] *means an average of four feature vectors concatenated with an average of two.*

3.5 Experiments with less teaching data

It would be interesting to know, whether the previous results could be still improved by increasing the amount of training data. Since no more data has been currently available, we tested the effect of reduced set of training data to see if it affects to the accuracies. The results show an approximate decrease of half percent, when two thirds of the original training data was used. A rough extrapolation of these results to the case, in which more training data would be available, gives hope for some improvements.

4 Continuous speech recognition

4.1 Information for segmentation

In this section we address the natural question that follows the previously described successful experiments: Although high accuracies can be obtained by classifying phonemes when their locations are known beforehand, are these context vectors applicable in combined segmentation and labeling, that is, transcribing speech into phoneme sequences?

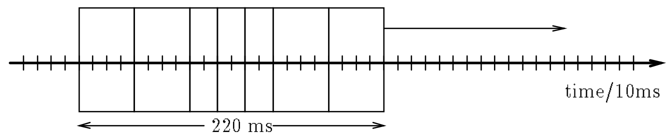

Figure 6: *The selected 140-dimensional context vector*

Our solution to this is to slide the time domain window (Fig. 6) over all of the the speech signal and to compute phoneme classification every 10 ms. This results in a so-called quasiphoneme sequence. The idea here is that since the context vectors cover such a long period of time domain, and the LVQ-codebook is still trained using phoneme centers only, transitions will be substantially different from stationary parts of phonemes resulting in high quantization errors. This error can then be used as an aid to segmentation. To develop this idea further, we have experimented with training another codebook for two classes only, for the phoneme centers and for transitions between phonemes. Figure 7 illustrates all this information available for segmentation.

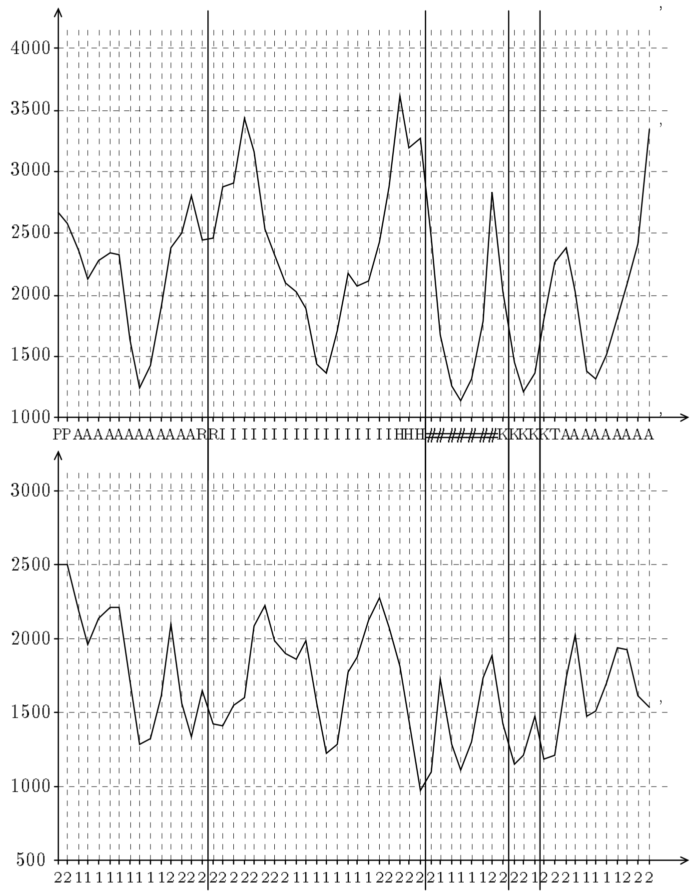

Figure 7: *Segmentation information for the Finnish word 'AIKA'. In the first graph the recognized quasiphonemes are shown together with their quantization errors. The lower graph shows the respective segmentation classifications. The real manual segmentations (not used in the experiments) are shown as vertical lines (the phonemes A,I,#,K ja A). The sign '#' denotes silence, '1' and '2' stand for stationary regions and transitions, respectively.*

4.2 HMM-Based Decoding

In our approach, the above mentioned classification results are interpreted as output symbols of a discrete-observation HMM. The details of the basic configuration can be found in [6]. The system first models each phoneme as a separate Markov source. A simple four-state left-to-right topology has been used. Output symbol distributions have been defined for state transitions within the phoneme models and they are trained using the standard maximum-likelihood approach [5].

Phoneme models are connected in parallel with empty transitions. The decoding algorithm is allowed to loop back from the final state of a phoneme model to the initial state of any other model. Probabilities for the transitions between phonemes were estimated separately from a large text corpus.

The main difference between a standard discrete observation (vector quantization) HMM–approach and our approach here is using an additional codebook for segmentation, and including the codebook quantization errors in the HMMs. For the quantization errors, both 8-level quantized values and continuos-density probability distributions with large constant variance were tried despite of the theoretical problems with requirements of statistical independence between codebooks.

Table 2 shows the final results of the experiments. It should be mentioned that the manually segmented starting and ending points of the spoken words are supposed to be known here, since we have not used RMS-values to detect them as in e.g. [6].

Method	correct%	errors%
HMM(STF)	93.86	9.50
HMM(FON)	97.07	4.58
HMM(FON, dQE)	97.03	4.14
HMM(FON, cQE)	97.64	3.78
HMM(FON, SEG)	98.43	3.15
HMM(FON, SEG, dQE)	98.09	3.15
HMM(FON, SEG, cQE)	98.43	2.94
HMM(FON/SEG)	98.50	3.02
HMM(FON/SEG, dQE)	98.04	3.18
HMM(FON/SEG, cQE)	98.49	2.84

Table 2: *Experimental results with different HMM-codebook variations. 'FON' stands for quasiphoneme codebook, 'SEG' segmentation-class codebook, 'dQE' and 'cQE' quasiphoneme quantization error codebooks in discrete and continuos case, respectively. 'FON/SEG' means that these sequences have been combined to form one larger codebook. For comparison, results obtained by using a short-time feature vector based quasiphoneme codebook ('STF') are given. The 'errors'-column includes all substitution, deletion and insertion errors.*

5 Conclusion

In this research we have applied the Learning Vector Quantization methods, including the latest developments [1, 3, 2, 4], to the task of Finnish speaker-dependent speech recognition. We experimented various ways to form a high-dimensional context vector by both averaging and concatenating short-time feature vectors within a time domain window. This method of taking the local phonemic context into account was shown to give excellent results when applied to the separate phoneme classification of Finnish speech. When compared to the case, in which one short-time feature vector was used, the classification accuracies increased from 87 % up to 99 % with the best form of context vector. It is also interesting to notice the positive effect of increasing LVQ-codebook sizes to the recognition results with high-dimensional context vectors.

We also applied the LVQ methods with context vectors to the segmentation task of Finnish speech and exploited this additional two-class time-series information successfully by using it as a separate HMM codebook or a combined codebook with the quasiphoneme sequence. We also tried to utilize the quantization errors, but they proved to give no significant improvement. When the starting and ending points of words were supposed to be known, the lowest error percentage was approximately 3 %. These figures show a significant improvement when compared to results obtained in real-time operation, based on separate classification of short-time feature vectors [6]. Naturally, the computing load is now multiplied, calling for massively parallel hardware.

References

[1] Teuvo Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin-Heidelberg-New York-Tokio, 3 edition, 1989.

[2] Teuvo Kohonen. Improved versions of learning vector quantization. In *Proceedings of the International Joint Conference on Neural Networks*, pages I 545–550, San Diego, June 1990.

[3] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[4] 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*, pages I 725–730, Baltimore, June 1992. IEEE.

[5] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[6] 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 the International Conference on Artificial Neural Networks (ICANN-91)*, pages 771–776, Espoo, Finland, June 24-28 1991.