International Conference on Spoken Language Processing (ICSLP'92), Banff, Canada, October 12-16, 1992, Vol. 1, pp. 551-554

Using Phoneme Group Specific LVQ-codebooks with HMMs

Pekka Utela, Samuel Kaski and Kari Torkkola†
Helsinki University of Technology
Laboratory of Information and Computer Science
Rakentajanaukio 2 C, Espoo SF-02150, FINLAND
tel: +358 0 451 3278, fax: +358 0 451 3277, email: Pekka.Utela@hut.fi
†IDIAP, CP 609, CH-1920 Martigny, SWITZERLAND, email: karit@idiap.ch

ABSTRACT

We have implemented a novel combination of LVQ–codebooks and hidden Markov models (HMMs) for transcribing spoken Finnish words into phonemes. A separate LVQ–codebook was trained for the unvoiced plosives /k/, /p/ and /t/. HMMs were then modified to accommodate two parallel phoneme codebooks and a third codebook representing the power of the speech signal. The error rate of phonemic transcription was as low as 9.1 %. A preliminary test with an additional /b/, /d/ and /g/ codebook was also carried out.

1 INTRODUCTION

LVQ: Ordinary vector quantizers replace an input vector with a discrete symbol corresponding to the closest codebook vector. Similar minimum–distance rule can also be used in pattern classification, if the symbols are made to represent different classes of data. Several codebook vectors may then belong to the same class. Although vector quantizers and minimum–distance classifiers are closely related, their design criteria are quite different.

Learning Vector Quantization (LVQ) [6] is an algorithm for optimizing minimum–distance classifiers. Before applying LVQ the codebook must be initialized by setting the initial values of the vectors and assigning them to classes. The algorithm can then be used to tune the codebook. The central idea of LVQ is to classify known samples and to make adjustments to the positions of codebook vectors according to the results. As this procedure is iterated the decision borders between classes are drawn towards the optimal Bayesian borders.

We have applied LVQ classifiers to phoneme recognition. Short-time feature vectors are first classified to phonemic symbols ("quasiphonemes"), which are then processed further to determine the actual phonemes. In the present version of our speech recognition system this basic method has been slightly modified: the input vectors are now constructed by concatenating several feature vectors from consecutive frames. The added information content has been found to improve recognition accuracy [4].

HMMs with LVQ: Discrete observation HMMs are usually applied to vector quantized data. The standard criterion in designing quantizer codebooks is to try to minimize distortion. Thus, the input data is represented as faithfully as possible. An alternative approach was studied by Iwamida et al. in [3]. They experimented with HMMs combined with an LVQ–trained codebook and achieved high accuracies in recognizing discrete Japanese phonemes. Comparisons with ordinary vector quantizers indicated, that LVQ clearly improves the discriminative power of HMMs. Iwamida et al. argued, that codebook distortion does not correlate trivially with recognition accuracy, as LVQ–tuning increased both accuracy and distortion in the chosen task.

Multiple codebooks with HMMs: Multiple parallel codebooks have earlier been used in quantizing different types of feature vectors, because with the specific codebooks it is possible to keep the overall quantization distortion reasonably low [2, 8]. Since each codebook produces a symbol in every frame, the models must be slightly modified. One alternative is the approach suggested in [8]. Multiple observations are assumed to be independent. The probability of the model emitting multiple symbols can then be computed as the product of the probability of producing each symbol.

Unvoiced plosive classification: During the pronunciation of the unvoiced plosives a complete closure is formed in a part of the vocal tract. In that period no sound is produced. At the end of the phoneme the closure is suddenly opened and the released air produces a burst sound, which lasts only a few centiseconds. Depending on the placement of the closure the spectral characteristics of the burst are modified by different parts of the vocal tract. Thus, the information differentiating these phonemes is concentrated in a short interval.

Discriminating between the unvoiced plosives /k/, /p/ and /t/ of Finnish language is a difficult task. One of the reasons for this is, that typical time–synchronous feature extraction does not capture burst sound properties optimally. Moreover, if the placement of codebook vectors in a quantizer depends on the input vector density, that is, the quantizer is designed for minimum distortion, the transient events are not well represented.

Previously, the problem was approached by training a separate classifier with the plosive bursts and the beginnings of words only [7]. The outputs of the two classifiers were combined with heuristic rules.

Phoneme group specific codebooks: By concentrating on a group of phonemes it was possible to design a codebook, which discriminated accurately between them. On the other hand, the multiple codebook approach with HMMs has suggested a way of combining distinct sources of information. In the present work we have integrated these two methods.

The main difference between our approach and ordinary multiple codebook HMMs is the criterion for designing the codebooks. Instead of minimizing quantization distortion, we optimized separate codebooks to discriminate phonemes of a smaller group. In this paper we study the properties of the system.

International Conference on Spoken Language Processing (ICSLP'92), Banff, Canada, October 12-16, 1992, Vol. 1, pp. 551-554

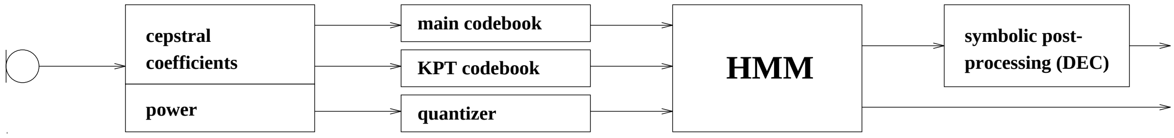

Figure 1: Block diagram of the system

2 THE SPEECH RECOGNITION SYSTEM

The system used in our experiments is a speaker dependent recognizer with unlimited vocabulary (also known as the "Phonetic Typewriter"). It was originally developed for Finnish language, but one version has also been tested with Japanese [7]. Phonemes are the basic units of recognition. This choice can be justified by the characteristics of the language: original Finnish contains only 21 phonemes, and nearly all of them are represented by single graphemes in the written form of the language.

Figure 1 illustrates the present configuration of the speech recognition system. A more detailed description of it, including a preliminary version of the multiple codebook HMMs, can be found in [10].

Audio signal is captured with a dynamic microphone and sampled at 12.8 kHz. The waveform is then blocked into 20 ms long frames overlapping by 10 ms. From these frames we compute 20 MEL–scaled cepstral coefficients [1] and signal power. Because the HMMs operate on discrete symbols, the power is quantized to 16 levels.

Two parallel codebooks classify concatenated feature vectors into phonemic symbols at 10 ms intervals. The streams of symbols and quantized power are then processed by the HMMs to determine a phonemic transcription of the utterance. Due to the nature of Finnish language this transcription nearly equals correct ortography.

The main source of errors at the phoneme level is coarticulation. In many cases these errors are quite systematic. Therefore it is possible to correct them (and other systematic errors) by context-dependent rules. *Dynamically Expanding Context (DEC)* is an algorithm, that derives the rules from examples of erraneous and correct strings [5]. This symbolic postprocessing method was not used in our experiments.

3 IMPLEMENTATION OF PHONEME RECOGNITION

Separate HMMs are trained for each phoneme and for the beginning and the end of an utterance. All models have the same 4-state left-right structure depicted in figure 2. The model emits multiple independent symbols with every transition.

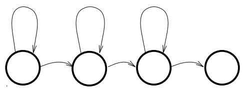

Figure 2: *A phoneme model*

Model parameters are estimated from prototype strings computed from manually segmented data. Standard forward-backward procedure is used in the estimation [9].

The phoneme models are then connected in parallel with empty transitions (i.e. transitions that do not produce output symbols). The models of the beginning and the end are also added to produce the word model presented in figure 3. The number of consecutive phonemes is not fixed, allowing words of different lengths to be decoded.

Probabilities for the transitions between phonemes were estimated separately from a large text corpus. However, these probabilities do not compare naturally with the joint probabilities of phoneme model transitions. Therefore we found it necessary to raise the values to a power, the value of the exponent depending on the number of parallel codebooks.

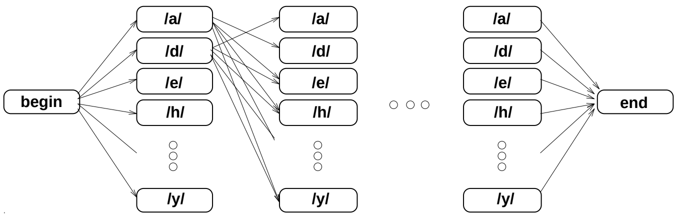

Figure 3: *The word model*

Symbol sequencies can be decoded by using Viterbi-search to find the most probable path through the word model. The phonemic transcription is then obtained by recording the phoneme models along this path.

4 EXPERIMENTS

4.1 Speech data

The speech material used in the experiments was collected from three male Finnish speakers. Each speaker repeated four times a set of 311 words (1737 phonemes). Speaker-dependent performance was evaluated by using three of these sets for training the system and one for testing. Four independent runs were made by using each set once as test material. Finally, an average of the results from all speakers and runs was calculated. Symbolic postprocessing was not used in the experiments.

4.2 Training of the codebooks

The main codebook was first initialized with the self-organizing map method [6] using all feature vectors from the training set. Thereafter, the codebook vectors were assigned to classes by majority voting. Samples used in the voting procedure were collected from the most stationary (middle) parts of the phonemes. The actual training of the codebook was done with LVQ1, again using only the most stationary parts.

4.3 A separate KPT codebook

The samples used in training a KPT codebook must be chosen carefully, while the information relevant to the classification of the unvoiced plosives lies in a short interval of time. The sharp rise of the power of the speech signal when the vocal tract opens serves here as a useful landmark. In pilot experiments it was found, that best results are obtained by concatenating four consecutive feature vectors together, starting a fixed time before this landmark. In order to gain some translation invariance, it was further found useful to pick three different samples from each unvoiced plosive, starting at successive timesteps. This procedure in effect slided a sampling window of length four over the signal in the region of the burst sound.

Because the transient behavior of the speech signal after a pause resembles that of the unvoiced plosives, also vectors representing the beginning of a word were included into the KPT codebook. The codebook was thus trained also with samples picked from the beginning points of all words, again by sliding a sampling window of length four.

Two versus three codebooks: Table 1 illustrates the differences of recognizing unvoiced plosives with a separate codebook and with the main codebook. With the former approach the results are clearly better, irrespective of the total number of vectors. This result does not allow straightforward inferences to be made about the benefits of a separate KPT codebook *per se*, however. In the two codebook case the main codebooks did not contain as many vectors dedicated to the unvoiced plosives as the KPT codebook did. Even in the codebook containing 500 vectors, only an average of 34 vectors represented the unvoiced plosives.

In the two codebook case, some experiments were also made with vectors representing the beginning of a word included in the main codebook. In this way it was made sure, that the differences in the results were not due to this difference in the codebooks. Also leaving these vectors out of the KPT codebook produced worse results as those reported below. The total sizes of the codebooks were the same in all of these experiments.

The effects of trying to differentiate the unvoiced plosives with the main codebook in addition to the separate codebook are also studied in table 1. Although the unvoiced plosives are indeed recognized better, the overall error rate is increased.

code-vectors	code-books	all phonemes		unvoiced plosives	
		correct %	errors %	correct %	errors %
280	3[2]	92.8	9.8	86.0	16.5
	2	91.0	11.4	75.0	27.7
315	3[1]	93.6	9.4	92.3	11.5
	3[2]	93.2	9.3	88.3	13.8
	2	91.2	11.3	75.2	27.7
356	3[2]	93.4	9.1	89.8	12.3
357	2	91.3	11.2	76.2	26.3
500	2	91.4	11.2	75.3	27.4

Table 1: *Phonemic transcription results obtained using two codebooks (the power and the LVQ–trained classifier) and by using a separate KPT codebook. In the three-codebook case the size of the main codebook was fixed to 216 units and only the size of the KPT codebook was varied. ([1] and [2]: The unvoiced plosives were and were not differentiated with the main codebook, respectively.)*

Besides using the vectors trained to classify unvoiced plosives as a third codebook, another possible approach is to add them to the main codebook. This destroys the optimality of the decision borders created by the LVQ-algorithms, but as can be seen in table 2, the percentage of recognition errors is only slightly greater than in the three–codebook case.

	all phonemes		unvoiced plosives	
	correct %	errors %	correct %	errors %
3 codebooks	92.9	9.7	87.1	15.2
2 codebooks	92.4	10.0	86.9	14.6
Difference	0.5	-0.3	0.2	0.4

Table 2: *Phonemic transcription results obtained by adding the vectors of the KPT codebook to the main codebook and using them as a third codebook with HMMs. The main codebook consisted of 216 and the transient codebook of 99 units. The length of all vectors corresponded to three feature vectors concatenated together.*

KPT codebook with a class for other phonemes: The auxiliary codebook might also contain a class representing all the phonemes that are discriminated in the main codebook. The new class would direct the HMMs to base the decisions on the symbols from the main codebook. It is, however, difficult to choose the training samples representing the other phonemes. We tried several approaches: collecting them from the stationary parts of the phonemes, from the regions where phonemes change, or quite randomly. The error rate was found to be clearly higher than with the KPT codebook described before. One possible reason to this finding may be, that the codebook vectors representing the beginning of a word already partly fulfilled the task intended for this new class. A clear majority of speech samples are quantized to belong to the beginning of a word by the KPT codebook.

The effects of changing the size of the KPT codebook: In order to find out more clearly, how a KPT codebook affects the recognition results, its size was varied while the size of the main codebook was kept fixed. As figure 4 indicates, changes in the size of the KPT codebook affect only the recognition errors of the unvoiced plosives.

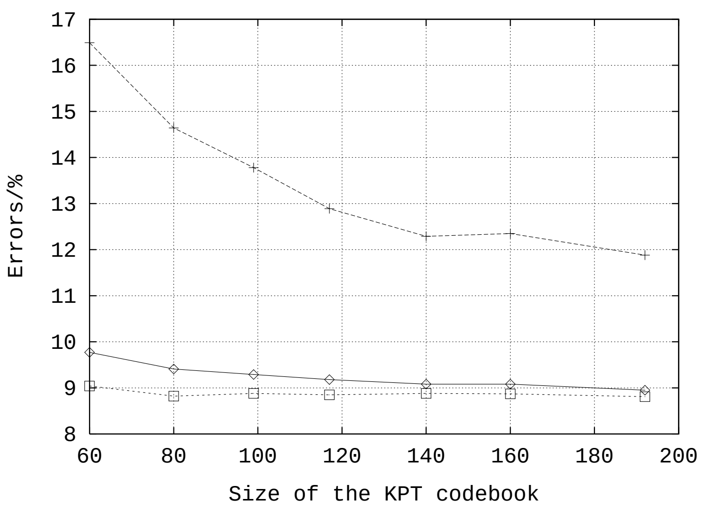

Figure 4: *Recognition errors when the size of the main codebook was fixed at 216 vectors, and the size of the KPT codebook was varied. Total errors are marked with oblique squares, errors in recognizing unvoiced plosives with "+"-signs and errors in recognizing other phonemes with right squares.*

4.4 Experiments with a BDG codebook

The voiced plosives /b/, /d/ and /g/ are also produced by forming a closure in the vocal tract and releasing it, but a periodic sound is emitted during the closure. Recognition of /b/ and /g/ had not been implemented earlier in the Finnish version of the system, because these phonemes do not exist in the original Finnish language. They do, however, appear in some words of foreign origin. In the Japanese version /b/, /d/ and /g/ were treated with a similar specific classifier as the unvoiced plosives [7].

We wanted to estimate the applicability of our method to recognizing the voiced plosives by executing a series of preliminary tests. A set of (Finnish) words containing these phonemes was prepared, and two repetitions were collected from one male speaker. Two independent tests were run in each case, using one set for training and the other for testing. The BDG codebook was trained with the transient parts of /b/, /d/, /g/ and the beginnings of words. Two samples, each consisting of five concatenated feature vectors, were picked near the rise of the power. The sizes of the main codebook, KPT codebook and BDG codebook were 216, 100 and 80 respectively.

Three comparisons were made. The first two versions used four codebooks: power, KPT, BDG and the main codebook. In the first experiment the voiced plosives were treated as a single class in the main codebook, while the second version had separate classes for them. The vectors used in training /b/, /d/ and /g/ in the main codebook were picked from the stationary (middle) parts of the phonemes. The last version distinguished voiced plosives in the main codebook and the BDG codebook was omitted.

	all phonemes		voiced plosives	
	correct %	errors %	correct %	errors %
4 codebooks[1]	85.8	19.6	77.8	29.9
4 codebooks[2]	88.1	16.7	89.5	12.6
3 codebooks	89.6	15.1	86.8	15.5

Table 3: *Phonemic transcription results with and without the BDG codebook. The first version with four codebooks (*4 codebooks[1]*) did not differentiate /b/, /d/ and /g/ in the main codebook, the second (*4 codebooks[2]*) did. Due to the small amount of training material and a different speaker, these results should not be compared with those in tables 1 and 2.*

The results suggest, that the recognition of the voiced plosives was based mainly on the stationary parts of these phonemes, since the error rate is clearly higher when the voiced plosives are not distinguished in main codebook (table 3, row 1). When /b/, /d/ and /g/ were separate classes in the main codebook, adding the BDG codebook increased the recognition accuracy of these phonemes (table 3, row 2). On the other hand, results with other phonemes suffered. Due to the limited amount of data the results are, however, only preliminary.

5 DISCUSSION

In summary, recognition results can be effectively improved by using a separate codebook trained with LVQ to classify unvoiced plosives. The improvement can be seen to concentrate on the recognition of the unvoiced plosives. The error rate was lowest with a separate KPT codebook, but the computationally less intensive approach of adding its vectors to the main one yielded only slightly higher error rate. In contrast to these findings, the preliminary experiments with a BDG codebook produced somewhat conflicting results.

The main advantage of employing an auxiliary codebook is that the discriminative power of LVQ can be focused on a selected subset of the different classes. Furthermore, with parallel codebooks it would be possible to use different input features for distinct groups of phonemes.

REFERENCES

[1] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980.

[2] V. N. Gupta, M. Lennig, and P. Mermelstein. Integration of acoustic information in a large vocabulary word recognizer. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP87)*, volume 2, pages 697–700, Dallas, Texas, April 1987.

[3] 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 (ICASSP90)*, volume 1, pages 489–492, Albuquerque, NM, USA, April 3-6 1990.

[4] J. Kangas. Phoneme recognition using time-dependent versions of self-organizing maps. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, May 14-17 1991.

[5] T. Kohonen. Dynamically Expanding Context, with application to the correction of symbol strings in the recognition of continuous speech. In *Proceedings 8th International Conference on Pattern Recognition (8th ICPR)*, pages 1148–1151, Paris, France, Oct. 27-31 1986.

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

[7] T. Kohonen, K. Torkkola, M. Shozakai, J. Kangas, and O. Ventä. Phonetic Typewriter for Finnish and Japanese. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP88)*, pages 607–610, New York City, USA, April 1988.

[8] K.-F. Lee. Large-vocabulary speaker-independent continuous speech recognition: the sphinx system. Technical Report CS-88-148, CMU, Apr. 1988.

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

[10] K. Torkkola, J. Kangas, P. Utela, S. Kaski, M. Kokkonen, M. Kurimo, and T. 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.