Proceedings of the Int. Conf. on Acoustics, Speech, and Signal Processing, Adelaide, Australia, April 19-22, 1994

NEW WAYS TO USE LVQ-CODEBOOKS TOGETHER WITH HIDDEN MARKOV MODELS

Kari Torkkola

IDIAP (Institut Dalle Molle D'Intelligence Artificielle Perceptive)
C.P. 609, CH-1920 Martigny, SWITZERLAND
email: karit@idiap.ch

ABSTRACT

We introduce a novel way to employ codebooks trained by Learning Vector Quantization together with hidden Markov models. In previous work, LVQ-codebooks have been used as frame labelers. The resulting label stream has been modeled and decoded by discrete observation HMMs. We present a way to extract more information out of the LVQ stage. This is accomplished by modeling the *class-wise quantization errors* of LVQ by continuous density HMMs. Experiments in a speaker dependent phoneme spotting task verify that significant improvements are attainable over plain continuous density HMMs, or over the hybrid of LVQ and discrete HMMs.

1 INTRODUCTION

Hidden Markov models (HMM) are today among the most succesful techniques in automatic speech recognition with well studied and mature training algorithms [12]. These techniques can be roughly divided into two main categories: continuous density HMMs (CDHMM) and discrete observation HMMs (dHMM) with semi-continuous (tied mixture) HMMs somewhere in between. Either continuous or discrete, the aim of the models is faithful *representation* of the feature vector sequence derived from the speech signal, either directly by mixtures of multivariate Gaussian or other distributions, or through vector quantization (VQ). Both the maximum likelihood training algorithms of the HMMs, and in the latter case, also the codebook construction algorithms aim at this.

In recent years many ways have been presented to enhance the discrimination capabilities of HMMs. These include, among others, new training criteria [17, 13, 5] and hybrids of discriminative methods with the HMMs. The latter type of systems have been recently dominated by artificial neural networks (ANN). These hybrids can further be grouped into two main clusters: ANNs as probability estimators for HMMs [1], or ANNs as codebooks or labelers [4, 15, 10, 9] for dHMMs.

The present paper is concerned with a novel combination falling into the latter category. The combination is based on *Learning Vector Quantization (LVQ)* [7, 8]. Many such LVQ/HMM-hybrids have been experimented with in the literature earlier [4, 19, 6, 15, 3, 10]. In these systems LVQ acts as a phonemic discriminative labeler. The resulting label stream is then modeled by dHMMs. This paper extends the previous work by presenting a way of extracting more information of the LVQ stage. This is founded on modeling the class-wise quantization error vectors of a LVQ codebook by using *continuous density* HMMs.

The structure of the rest of this article is as follows. Section 2 gives an overview of LVQ and reviews its prevalent use together with HMMs. In Sec. 3 we present a way of preserving more information of the LVQ stage, and suggest CDHMMs as the tool process that information. In Sec. 4 we describe our experiments in speaker-dependent phoneme spotting task, and Sec. 5 is devoted to discussion.

2 LVQ-CODEBOOKS IN SPEECH RECOGNITION

The role of conventional vector quantization algorithms in speech recognition, such as the Linde-Buzo-Gray algorithm, or K-means, is to *represent* speech feature vectors with the smallest possible distortion. This is not the case with the Learning Vector Quantization (LVQ) methods, which try to aim at *discrimination* of pattern classes, whatever they may be [7, 8]. Codebook vectors directly define the class borders in the feature space according to nearest-neighbor rule. LVQ modifies the codebook vectors adaptively so that the borders between classes will approximate Bayes' decision surfaces. Quantization error (distortion) is of secondary interest.

In literature, the customary practice of using LVQ is as a substitute to conventional vector quantization. Short time feature vectors are used as the basis for classification, most commonly to phoneme classes. HMMs are then used to combine local classification decisions by treating them just as VQ-codebook indices are treated with dHMMs. In phoneme-related tasks, training has been done either by single frames [6], or by concatenating several frames together to represent some context [19, 4, 15, 3]. Further, the whole (phoneme) token has been used in training. All of the frames or successive shifts of the concatenated frames have acted as examples. However, this might pose a problem as short-time segments of the speech signal close to transitions between phonemes might be extremely confusing and

might resemble very much other parts of other phonemes.

When the task has been speaker dependent and phoneme oriented, significant improvements have been observed due to LVQ when compared against ordinary VQ [4, 15]. Where the task has been word or phrase recognition, the results have been slightly controversial: no improvement in [19], but significant improvements in [3].

In a recent study it was shown that by taking the local context into account (as much as 220 ms) in the construction of the pattern vectors, performance can be significantly increased compared to using a single frame or a few concatenated frames [10]. Codebooks order of magnitude larger than conventionally used are needed to represent variations in these high dimensional "context vectors". Another LVQ-codebook, whose purpose is to discriminate phoneme centers from transitions, was introduced in parallel with the main codebook.

In addition to enhanced discriminative properties, another advantage of the LVQ over conventional VQ is that the discrete alphabet in phonetically motivated classification (i.e., the number of classes) is considerably smaller than the number of codebook vectors in usual VQ. This results in less output probability parameters to be estimated for the dHMMs. Complicated smoothing schemes are thus usually unnecessary.

3 CLASS-WISE QUANTIZATION ERRORS

3.1 Extracting More Information from the LVQ

All previous work on LVQ-codebooks in speech recognition has concentrated on using the *class label of the closest codebook vector* only. To decode this label stream, discrete observation HMMs have been employed.

However, the normal practice with pattern classifiers, extracting only the final decision of the classification (the class label only) is desirable only when that really is the final decision stage of the whole task. This is not the case with the LVQ/HMM hybrids in speech recognition; the final decision is made by the Viterbi search at the HMM-stage. It is neither optimal nor necessary to resort to too early hard decisions at this point.

Remaining with dHMMs, an obvious improvement could be to preserve information about several of the closest classes, possibly weighted in some manner. Some kind of heuristics is required to convert the distance of the current feature vector to the closest classes into usable weights or probabilities. Techniques of fuzzy VQ might be helpful for this stage. Certainly more information would be acquired from the same amount of training data. Alternatively, less training data would result in the same performance, as at each time step, not only the probability corresponding to the best label, but many or all of them will be gaining information.

However, the idea that we pursue here, is to make use of the actual measure that is the basis of LVQ-classification, the Euclidean distances to the closest codebook vectors of each class. Although these distances could be transformed into weights or probabilities, LVQ is not really designed keeping probability estimation in mind.

A step in this direction was already taken in [10]. In addition to the classification label stream provided by LVQ, also the VQ error was exploited. VQ error refers now to the conventionally employed scalar quantity: distance to the closest codebook vector, regardless of its label. In contrast to other related work cited in Sec. 2, LVQ was trained by context windows positioned at phoneme centers only. Since the input representation was constructed by taking into account a relatively long duration of context, phoneme borders appear very different from the centers resulting in high quantization errors at transitions between phonemes. The quantization error can thus both give indication of the positions of the phonemes, and their closeness to the corresponding codebook vector. Multiple input stream HMMs were used to combine the two sources of information, one discrete, another continuous valued.

In this paper, we extend the above work by discarding the label of the closest class. Instead, the distances to codebook vectors of different classes, that can also be called as class-wise quantization errors, can be used directly without resorting to any transformations to weights or probabilities. *Continuous observation density* HMMs can then be applied to model a stream of these vectors. In computation this contains no more work than normal vector quantization. Only some more bookkeeping is needed to store the closest distances for each of the classes. However, the HMM-part will consist of the normal continuous density HMMs instead of faster and simpler dHMMs. The proposed configuration thus combines computationally intensive parts of both VQ/dHMM-, and CDHMM-systems. However, significantly improved recognition accuracies can be achieved with the configuration, as will be described in Sec. 4.

3.2 LVQ as a Feature Transformation

In pattern recognition one is always concerned with choosing or computing such a representation of the input data, that enhances class separability while keeping within-class variability as small as possible. Several linear transformations, that are optimal according to criteria usually connected to between-class and within-class scatter matrices can be found in pattern recognition textbooks, and they have been applied succesfully to speech recognition [2].

In the suggested scheme, LVQ can be viewed as a nonlinear, discriminative feature transformation before the CDHMM-stage. A simpler version of such a transformation has been presented for example in [14, pp. 85-87], where a new feature vector is formed by computing distances to fixed points in the feature space. These points represent the means of pattern classes, while in case of LVQ, there are several such points per class, and their locations been chosen so as to minimize classification errors through LVQ-training. Other kinds of non-linear feature transformations that are derived on the basis of training by examples have also been used in speech recognition [16].

If the intent is to make a classification decision based on a single feature vector it does not, of course, make any sense to regard LVQ as a feature transformation. LVQ can then just simply do the decision. However, when the feature vectors are parts of a larger entity and when a decision has to be made taking all parts into account, as in speech recognition, this might be a useful point of view.

4 EXPERIMENTS

We will now describe experiments to compare the performance of the suggested method with some established algorithms. A comparison between four architectures is presented in Table 1. The architectures are:

1. conventional CDHMMs modeling a stream of cepstral vectors,
2. LVQ/dHMM-hybrids, where LVQ produces a stream of class labels, which is modeled by discrete observation HMMs, and
3. the new approach proposed in this paper, where LVQ produces a stream of class-wise quantization error vectors, which is modeled by CDHMMs, and
4. parallel use of the class-wise quantization error with the cepstral vectors.

The task in this comparison is *speaker-dependent phoneme spotting* in the Finnish language [15, 10]. The database contains four repetitions of a set of 311 utterances spoken by three male Finnish speakers. Each set consists of 1737 phonemes. In the original Finnish language, there are only 21 different phoneme classes: 8 vowels and 13 consonants. Four additional phonemes have been adopted with loan words from other languages, but none of those were represented in the database. There were thus 22 phonemic classes for the LVQ to differentiate (21 phonemes and silence), which was also the dimensionality of the class-wise quantization error vectors. Three of the repetitions were used each time for training, and the remaining one for testing. Four independent runs were made for each speaker by leaving one set at a time for testing. Thus all recognition results presented in this paper are speaker-dependent, averages of 12 test runs, and based on 20844 phoneme spotting scores.

Speech analysis conditions were the following: 12.8 kHz sampling rate, pre-emphasis coefficient 0.95, 25.6 ms Hamming window every 10 ms, and 20-component mel-scale cepstral coefficients (MFCC) computed for every window. Where difference coefficients (Δ) have been used, they were computed from a period of 40 ms. This applies also to the quantization error vectors.

For context vectors as LVQ input, the codebook size was 2000, and for the MFCC, it was 500. LVQ-training procedure was exactly the same as described in [10]: using only phoneme centers, initialization by K-nearest-neighbor, OLVQ1 for 10000 iterations with $\alpha(0) = 0.3$, and LVQ1 using $\alpha(0) = 0.2$ In the latter stage, the number of iterations was 100 times the number of codebook vectors.

Looking at the results in Table 1, it is obvious that exploiting all class-wise quantization errors instead of using only the class identity of the closest codevector preserves more information for the latter decision stages. This is reflected as the difference between LVQ/dHMM-hybrid and LVQ/CDHMM-hybrid. We can also see clearly, that the class-wise quantization errors are very interdependent, since modeling them as 22 separate streams, or as a single stream with diagonal covariances produces poorer results. We got our best results by using full covariance matrices, and two Gaussian mixtures.

Though the LVQ/CDHMM-hybrid is better than plain CDHMMs, the difference is barely significant. On the other hand, the LVQ+MFCC/CDHMM-hybrid beats plain CDHMMs by a clear margin. Due to a relatively large number of tests, the confidence limits (99%) are relatively tight: $\pm 0.23\%$ for the best result (1.8%). This enables us to state that the proposed architecture is significantly better than a LVQ/dHMM-hybrid (4.6%), or a CDHMM system (2.9%) in this task. In the case of plain CDHMMs, using second order derivatives, increasing the number of mixtures, or using diagonal covariance matrices with larger number of mixtures did not improve our best CDHMM result.

Combination	LVQ Input	HMM Input	Number of HMM input streams	Mixtures/ stream	Covariance	Error rate
1	-	MFCC+Δ	2	5+3	diag	3.2
	-	MFCC+Δ	2	2+1	full	2.9
2	MFCC	LVQ-best label	1	-	-	8.5
	220 ms of context	LVQ-best label	1	-	-	4.6
3	220 ms of context	LVQ-qerr+Δ	2	7+7	diag	5.4
	220 ms of context	LVQ-qerr+Δ	2	2+1	full	2.5
	220 ms of context	LVQ-qerr	22	5	diag	16.8
	MFCC	LVQ-qerr+Δ	2	2+1	full	4.9
4	220 ms of context	MFCC+Δ+LVQ-qerr+Δ	4	2+1+2+1	full	1.8

Table 1. Experimental comparison between CDHMMs (1) LVQ/dHMM-hybrids (2), and LVQ/CDHMM-hybrids (3, 4). "MFCC" refers to 20 component mel-scale cepstral vectors. Δ refers to difference coefficients. "220 ms of context" denotes the context vector described in [10]. "qerr" refers to class-wise quantization errors computed from the LVQ-stage.

In the LVQ/dHMM-hybrid we used only the label sequence produced by LVQ; not any other information, as suggested in [10]. Including another LVQ-codebook for phoneme center/transition classification, or the whole codebook quantization error, would no doubt improve the performance of the LVQ/dHMM-hybrid, as it did in [10]. However, nothing prevents us from using all this information in any free combination, employing multiple input stream HMMs, where some of the streams can be discrete and some continuous at the same time. We are currently investigating such combinations.

It must be noted that throughout this comparison, exactly the same training conditions and algorithms (embedded Baum-Welch training) have been used. The basic phoneme model structure has also been the same throughout all the experiments (3 emitting states/model, no skips). The comparison is thus actually made between different input representations, and not, for example, between different HMM-software packages. In all of the experiments we used a public domain software package LVQ_PAK [8], and a commercial package HTK [18] for the HMMs. A discrete observation version of the HTK was written for dHMM-experiments.

5 CONCLUSION

We have reviewed ways of employing LVQ-based codebooks with HMMs in speech recognition. We wanted to point out that the prevalent practice in the literature, using LVQ as a frame labeler, and modeling the label stream with dHMMs, unnecessarily makes too early hard decisions. We have demonstrated that modeling class-wise quantization errors by CDHMMs leads to significantly better results, and can even be better than the mainstream HMM-techniques. Using VQ-error as an indication of phoneme locations and its use as an input stream to HMMs was introduced in [10], to which this work is a direct extension. Other ways to employ VQ-error with HMMs have been presented, for example, in [11].

The presented scheme might also resemble superficially semi-continuous HMMs (SCHMM),but that is not the case. SCHMMs also utilize a "codebook", but it is a codebook of (Gaussian) distributions without any concept of classes. SCHMMs try to *represent* the current feature vector by making use of *a number of the closest codebook entries*, while the presented hybrid uses *for every class the codebook entry closest to the current feature vector*. It would not make sense to take into consideration the second or third closest codebook entries of a class even though they would be closer than the closest entries of other classes, since LVQ is trained in view that only the closest one matters.

REFERENCES

[1] H. Bourlard, N. Morgan, and S. Renals. Neural nets and hidden Markov models: Review and generalizations. *Speech Communication*, 11:237–246, 1992.

[2] M. J. Hunt and C. Lefebvre. A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. In *Proc. ICASSP*, pages 262–265, Glasgow, Scotland, May 23-26 1989.

[3] H. Iwamida, S. Katagiri, and E. McDermott. Speaker independent large vocabulary word recognition using an LVQ/HMM hybrid algorithm. In *Proc. ICASSP*, pages 553–556, Toronto, Canada, May14-17 1991.

[4] H. Iwamida, S. Katagiri, E. McDermott, and Y. Tohkura. A hybrid speech recognition system using HMMs with an LVQ-trained codebook. In *Proc. ICASSP*, pages 489–492, Albuquerque, NM, USA, April 3-6 1990.

[5] S. Kapadia, V. Valtchev, and S. J. Young. MMI training for continuous phoneme recognition on the TIMIT database. In *Proc. ICASSP*, pages 491–494, Minneapolis, MN, USA, April 27-30 1993.

[6] D. G. Kimber, M. A. Bush, and G. N. Tajchman. Speaker-independent vowel classification using hidden Markov models and LVQ2. In *Proc. ICASSP90*, pages 497–500, Albuquerque, NM, USA, April 3-6 1990.

[7] T. Kohonen, G. Barna, and R. Chrisley. Statistical pattern recognition with neural networks: Benchmarking studies. In *Proc. ICNN*, pages 61–68, San Diego, July 1988.

[8] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola. LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms. In *Proc. IJCNN*, pages 725–730, Baltimore, June 1992.

[9] P. LeCerf and D. V. Compernolle. Using parallel MLPs as labelers for multiple codebook HMMs. In *Proc. ICASSP*, pages 561–564, Minneapolis, MN, USA, April 27-30 1993.

[10] J. Mäntysalo, K. Torkkola, and T. Kohonen. LVQ-based speech recognition with high-dimensional context vectors. In *Proc. ICSLP*, pages 539–542, Banff, Alberta, Canada, Oct. 12-16 1992.

[11] S. Nakagawa and H. Suzuki. A new speech recognition method based on VQ-distortion measure and HMM. In *Proc. ICASSP*, pages 676–679, Minneapolis, MN, USA, April 27-30 1993.

[12] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[13] D. Rainton and S. Sagayama. Appropriate error criterion selection for continuous speech HMM minimum error training. In *Proc. ICSLP*, pages 233–236, Banff, Alberta, Canada, Oct. 12-16 1992.

[14] C. W. Therrien. *Decision, Estimation, and Classification.* John Wiley and Sons, 1989.

[15] K. Torkkola, J. Kangas, P. Utela, S. Kaski, M. Kokkonen, M. Kurimo, and T. Kohonen. Status report of the Finnish phonetic typewriter project. In *Artificial Neural Networks (Proc. ICANN)*, pages 771–776, Espoo, Finland, June 24-28 1991. North-Holland.

[16] V. Valtchev, S. Kapadia, and S. J. Young. Recurrent input transformations for hidden Markov models. In *Proc. ICASSP*, pages 287–290, Minneapolis, MN, USA, April 27-30 1993.

[17] S. Young. Competitive training: a connectionist approach to discriminative training of hidden Markov models. *Proc. IEE*, 138:61–68, 1991.

[18] S. J. Young. *HTK: Hidden Markov model toolkit V1.4 – Reference manual.* Cambridge University Engineering Department, October 1992.

[19] G. Yu, W. Russel, R. Schwartz, and J. Makhoul. Discriminant analysis and supervised vector quantization for continuous speech recognition. In *Proc. ICASSP*, pages 685–688, Albuquerque, April 3-6 1990.