Proc. of the IEEE Workshop on Neural Networks and Signal Processing, Ermioni, Greece, Sept. 6-8, 1994.

LVQ as a feature transformation for HMMs

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

Abstract. We present a new way to take advantage of the discriminative power of Learning Vector Quantization in combination with continuous density hidden Markov models. This is based on viewing LVQ as a non-linear feature transformation. *Class-wise quantization errors* **of LVQ are modeled by** *continuous density HMMs*, **whereas the practice in the literature regarding LVQ/HMM hybrids is to use LVQ-codebooks as frame labelers and discrete observation HMMs to model a stream of such labels. As decision making at frame level is suboptimal for speech recognition, the presented method is able to preserve more information for the HMM stage. Experiments in both speaker dependent and speaker independent phoneme spotting tasks suggest 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 among the most succesfull techniques in automatic speech recognition with well studied and mature training algorithms [14]. These techniques can be roughly divided into two main categories: continuous observation density HMMs (CHMM) 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 discrete 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 [18, 2, 7] 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, 3], or ANNs as codebooks or labelers [6, 16, 12, 11] 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)* [9, 10]. In the literature, many such LVQ/HMM-hybrids have been presented [5, 6, 8, 12, 16, 20], in which 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 followed by *continuous density* HMMs. This is founded on viewing LVQ as a feature transformation.

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 CHMMs as the tool process that information. In Sec. 4 we describe our experiments in two phoneme spotting tasks, 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 [9, 10]. 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 act as the basis for classification, most commonly to phoneme classes. HMMs are then employed 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 [8], or by concatenating several frames together to represent some context [20, 6, 16, 5]. 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 [6, 16]. Where the task has been word or phrase recognition, the results have been slightly controversial: no improvement in [20], but significant improvements in [5].

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 [12]. 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 smaller than the number of codebook vectors in usual VQ (few tens as opposed to few hundreds). This results in an oder of magnitude smaller amount of 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) 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 suboptimal to resort to too early hard decisions at the frame level.

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 output 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 [12]. 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 of 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 (class-wise quantization errors), can be used directly without resorting to any transformations to weights or interpretations as probabilities. *Continuous observation density* HMMs can then be applied to model a stream of these vectors. Fig. 1 illustrates the computation of the new feature vectors.

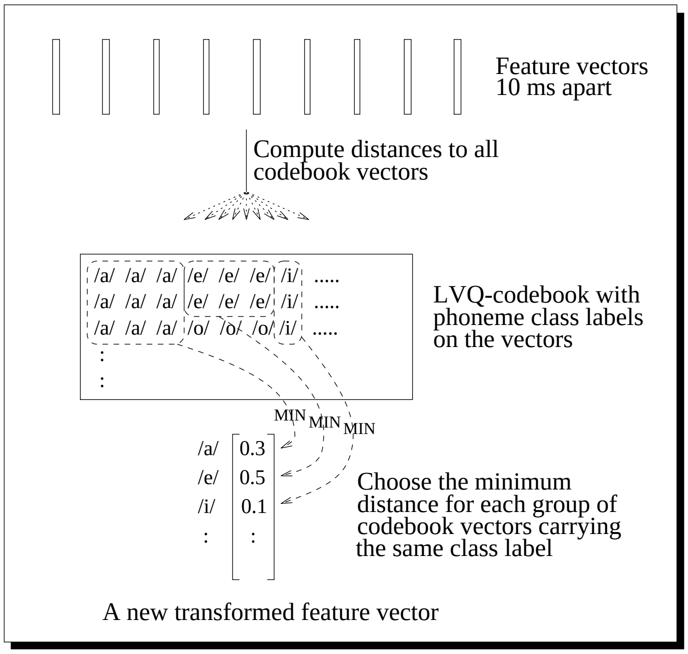

Figure 1: An illustration of the feature transformation. The stream of resulting new feature vectors is then modeled by HMMs.

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 [4].

In the suggested scheme, LVQ can be viewed as a nonlinear, discriminative feature transformation before the CHMM-stage. A simpler version of such a transformation has been presented for example in [15, 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 [17].

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 is a useful point of view.

3.3 Relation to semi-continuous HMMs

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.

4 Experiments

4.1 Speaker dependent phoneme spotting

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 CHMMs 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 CHMMs, and
4. parallel use of the class-wise quantization error with the cepstral vectors.

combi-nation	LVQ Input	HMM Input	stre-ams	mixts/ stream	covar	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
	context	LVQ-best label	1	-	-	4.6
3	context	LVQ-qerr+Δ	2	7,7	diag	5.4
	context	LVQ-qerr+Δ	2	2,1	full	2.5
	MFCC	LVQ-qerr+Δ	2	2,1	full	4.9
4	context	MFCC+Δ+LVQ-qerr+Δ	4	2,1,2,1	full	1.8

Table 1: Comparison between CHMMs (1), LVQ/dHMM-hybrids (2), and LVQ/CHMM-hybrids (3, 4) in a speaker-dependent task. “MFCC” refers to 20 component mel-scale cepstral vectors. Δ refers to difference coefficients. “context” denotes the 220 ms context vector described in [12]. “qerr” refers to class-wise quantization errors computed from the LVQ-stage. In all cases the HMMs had 3 emitting states.

The task in this comparison is *speaker dependent phoneme spotting* in the Finnish language [12, 16]. 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 speaker dependent recognition results presented in this paper are 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. The context vectors had a time span of 220 ms [12]. LVQ-training procedure was exactly the same as described in [12]: 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 LVQ1 stage, the number of iterations was 100 times the number of codebook vectors. An example of the output of the LVQ stage is presented in Fig. 2. Phoneme center locations can be clearly distinguished.

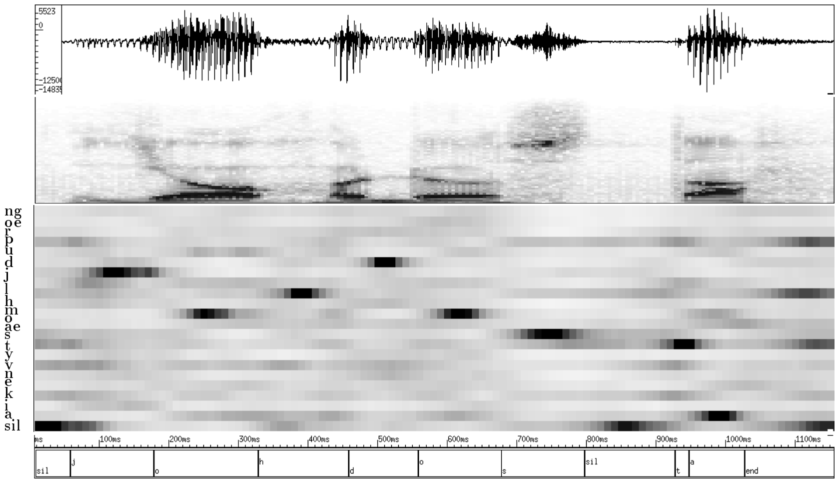

Figure 2: An example of the output of the LVQ-stage. From top to bottom: the speech signal (a Finnsh word */johdosta/*), its spectrogram, class-wise quantization errors with the phoneme labels on the left (darker shade denotes small values), and the labeling of the utterance.

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/CHMM-hybrid. We can also see that the class-wise quantization errors are interdependent, since modeling them by diagonal covariances produces poorer results. We got our best results by using full covariance matrices, and two Gaussian mixtures.

Though the LVQ/CHMM-hybrid is better than plain CHMMs, the difference is barely significant. On the other hand, the LVQ+MFCC/CHMM-hybrid beats plain CHMMs 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 CHMM system (2.9%) in this task. In the case of plain CHMMs, using second order derivatives, increasing the number of mixtures, or using diagonal covariance matrices with larger number of mixtures did not improve our best CHMM result.

In the LVQ/dHMM-hybrid we used only the label sequence produced by LVQ; not any other information, as suggested in [12]. 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 [12].

4.2 Speaker independent phoneme spotting

In this experiment, our aim was to find out whether this LVQ/HMM hybrid is applicable to the speaker-independent case. The database consisted of Swiss-French telephone speech with 56 speakers (about 2 hrs of speech). Half of the speakers were used for training and the other half for testing. The vocabulary also varied across the speakers.

Speech was sampled at 8 kHz, and 12-component mel-scale cepstra were computed each 10 ms. As the input to LVQ, we used slightly narrower context windows whose duration was 140 ms. The LVQ codebook size remained as 2000. 36 context-independent HMM phoneme models were used with 4 emitting states and full covariances throughout this experiment.

combination	HMM input	streams	mixts/ stream	correct	accuracy
1	MFCC+Δ	2	2,1	60.9	53.4
	LVQ-qerr+Δ	2	2,1	61.4	55.8
3	LVQ-qerr+Δ+$\Delta\Delta$	3	1,1,1	63.5	57.4
	MFCC+Δ+LVQ-qerr+Δ	4	2,1,2,1	63.9	58.0
	MFCC+Δ+LVQ-qerr+Δ+$\Delta\Delta$	5	1,1,1,1,1	65.3	59.4
4	MFCC+Δ+LVQ-qerr+Δ+$\Delta\Delta$	5	3,1,1,3,1	65.7	60.0
	MFCC+Δ+(LVQ-qerr)+Δ+$\Delta\Delta$	4	3,1,(1),3,1	67.7	61.7

Table 2: Comparison between CHMMs (1) and LVQ/CHMM-hybrids (3, 4) in a speaker independent task.

The gain of using LVQ in this (much harder) task is not as dramatic as in the first one, but comparing the baseline CHMM recognizer (the first row) to the result on the last row we can see that the improvement is anyway very significant. In addition to Δ-coefficients we also tried 2nd oder difference coefficients for the quantization error stream, which turned out to be advantageous. In addition, it seems that the actual quantization errors are less important than their difference and 2nd order difference coefficients. In the result of the last row of the table we dropped that altogether out but kept the Δ- and $\Delta\Delta$-streams, and the results improved.

It must be noted that throughout these comparisons, 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 each experiment. 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 [10], and a commercial package HTK [19] 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 CHMMs 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 [12], to which this work is a direct extension. Other ways to employ VQ-error with HMMs have been presented, for example, in [13].

Acknowledgement

The author would like to thank Mr. Albert Klaseboer for running experiments with the Swiss-French database.

References

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

[2] W. Chou, B. H. Juang, and C. H. Lee. Segmental GPD training of HMM based speech recognizer. In *Proc. ICASSP*, pages 473–476, San Francisco, CA, USA, March 23-36 1992.

[3] C. Dugast, L. Devillers, and X. Aubert. Combining TDNN and HMM in a hybrid system for improved continuous-speech recognition. *IEEE Trans. on Speech and Audio Processing*, 2:217–223, 1994.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] P. Le Cerf, W. Ma, and D. Van Compernolle. Multilayer perceptrons as labelers for hidden Markov models. *IEEE Trans. on Speech and Audio Processing*, 2:185–193, 1994.

[12] J. Mäntysalo, K. Torkkola, and T. Kohonen. Mapping context dependent acoustic information into context independent form by LVQ. *Speech Comm.*, 14:119–130, 1994.

[13] 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.

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

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

[16] 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.

[17] 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.

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

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

[20] 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, NM, USA, April 3-6 1990.