A NEW CODEBOOK TRAINING ALGORITHM FOR VQ-BASED SPEAKER RECOGNITION

Jialong He, Li Liu, and Günther Palm
Abteilung Neuroinformatik, University of Ulm
89069 Ulm, Germany
jialong@neuro.informatik.uni-ulm.de

ABSTRACT

VQ-based speaker recognition has proven to be a successful method. Usually, a codebook is trained to minimize the quantization error for the data from an individual speaker. The codebooks trained based on this criterion have weak discriminative power when used as a classifier. The LVQ algorithm can be used to globally train the VQ-based classifier. However, the correlation between the feature vectors is not taken into consideration, in consequence, a high classification rate for feature vectors does not lead to a high classification rate for the test sentences. In this paper, a heuristic training procedure is proposed to retrain the codebooks so that they give a lower classification error rate for randomly selected vector-groups. Evaluation experiments demonstrated that the codebooks trained with this method provide much higher recognition rates than that trained with the LBG algorithm alone, and often they can outperform the more powerful Gaussian mixture speaker models.

1. INTRODUCTION

Vector quantization (VQ) based speaker recognition is a conventional and successful method [1]. The basic idea in this approach is to compress a large number of short-term spectral vectors into a small set of code vectors. A codebook can also be viewed as a generalization of the long-term average where the short-term spectral variations due to different textual content are not averaged out but are modeled by separate code vectors. The successful modeling of the underlying acoustic classes allows the VQ-based system to achieve high recognition accuracy even with very short test utterances. A VQ codebook is usually trained with the LBG algorithm [2] to minimize the quantization error when replacing all feature vectors with their corresponding nearest code vectors. The codebook trained based on above criterion tends to represent the density and clustering of the training data. One weakness of using the LBG codebooks as a classifier is its weak discriminative power due to the fact that only the samples within a class, but no competitive data, have been used during the training process. In other words, minimizing the quantization error of the codebook does not necessarily lead to an optimal classification performance.

Several algorithms have been proposed by Kohonen [3] to globally optimize the codebooks after they are generated with a certain unsupervised learning algorithm. These algorithms are called learning vector quantization (LVQ). Instead of seeking an optimal approximation to the density functions of the training samples, the codebooks trained with one of the LVQ algorithms tend to define directly the classification borders between classes according to the nearest-neighbor rule. However, the classification decision for a speaker depends on the vector sequence derived from a test sentence rather than on an individual vector, thus, a higher correct classification rate for feature vectors achieved with the LVQ codebooks does not necessarily lead to a higher speaker identification rate. This is because the feature vectors are highly correlated, and this correlation has not been taken into consideration in the LVQ algorithm. In this paper, we develop a new supervised learning procedure so that the codebooks are trained to give a lower classification error rate for randomly selected vector-groups. Since this optimization criterion is consistent with the speaker classification decision rule, the codebooks trained with this method provide much higher speaker identification rates than that trained with the LBG or the LVQ algorithms, and can even outperform the more powerful Gaussian mixture speaker models (GMMs).

This work is partially supported by the state of Baden Württemberg, Germany (Landesschwerpunkt Neuroinformatik)

2. ALGORITHM

Suppose that there are L speakers and their codebooks $\{Y_k \quad k=1,2,...,L\}$ have already been generated by the LBG algorithm. During the test, an unknown speaker provides a test sentence from which a set of test vectors, $\{\vec{x}_t\}_1^T$, can be derived using the short-term analysis techniques. The distortion of a single vector $\vec{x}_t$, when quantized using the codebook Y_k, is given as

$$s_k(\vec{x}_t) = \min_{\vec{y}_j^k \in Y_k} \left[d\left(\vec{y}_j^k, \vec{x}_t\right) \right] \tag{1}$$

and the average distortion of the whole test sentence is

$$S_k = \frac{1}{T}\sum_{t=1}^{T} s_k(\vec{x}_t) \qquad 1 \le k \le L \tag{2}$$

then, the unknown speaker is identified as the reference speaker whose model gives the smallest quantization error

$$ID = \arg\min_{1\le k\le L}\{S_k\} \tag{3}$$

Obviously, the classification decision for an unknown speaker is made depending on the scores from all models. In contrast, the LBG codebooks are trained individually. We believe it is possible to further reduce the classification error by employing a proper discriminative training algorithm. The following iteration procedure can be applied after the codebooks are initialized by the LBG algorithm.

1. Randomly choose a speaker, denoted as speaker *j*.
2. From the training data of speaker *j*, select *N* vectors $\{\vec{x}_t\}_1^N$ as a vector-group.
3. Calculate the average distortion of these *N* vectors from each speaker's model using Eq. (2), now *T*=*N*. If the following conditions are satisfied, then go to step 4, otherwise go to step 5.
   (a) S_i is the smallest value but $i \neq j$,
   (b) $\left(S_j - S_i\right) / S_j < w$, where *w* is the window size.
4. To each vector $\vec{x}_t$, suppose the nearest code from the codebook of speaker *j* is $\vec{y}_m^j$, and the nearest code from the codebook of speaker *i is* $\vec{y}_n^i$, adjust these two code vectors as follows

$$\vec{y}_m^j \Leftarrow \vec{y}_m^j + \alpha\left(\vec{x}_t - \vec{y}_m^j\right)$$

$$\vec{y}_n^i \Leftarrow \vec{y}_n^i - \alpha\left(\vec{x}_t - \vec{y}_n^i\right)$$

   where α is a learning rate, after all vectors in the current vector-group being processed, go to step 1.
5. In the case that the current vector-group is correctly classified, for each vector $\vec{x}_t$, suppose $\vec{y}_m^j$ is the nearest code vector in speaker *j*'s codebook, adjust $\vec{y}_m^j$ by

$$\vec{y}_m^j \Leftarrow \vec{y}_m^j + \varepsilon\alpha\left(\vec{x}_t - \vec{y}_m^j\right)$$

   where α is the same learning rate as in step 4 and ε is another small constant to scale down the learning rate. After all vectors in the current vector-group have been processed, go to step 1.

The above iteration procedure can be repeated until the given iteration number is reached. It is easy to see that after the modification in step 4, the average distortion for the current misclassified vector-group from the correct speaker model decreases and that from the wrong speaker model increases. Similar to the LVQ3 algorithm, step 5 intends to keep the codebooks approximating the distributions of the training data. In fact, the above learning procedure can be regarded as an extension of the LVQ algorithm. If the number of vectors in each vector-group is set to one (*N*=1), then the training procedure is aimed at reducing the number of misclassified vectors, which is exactly the goal of the LVQ algorithm. Therefore, we named the above algorithm as the Group Vector Quantization (GVQ) algorithm.

The three parameters, learning rate (α), window size (*w*) and the constant (ε), have the same meaning as that described in the LVQ3 algorithm. An extra parameter to be determined is the number of vectors in each vector-group (*N*). If the goal of the training procedure is to reduce the number of misclassified vectors, then *N* should be set to 1. On the other hand, if we want to design the codebooks for speaker identification, *N* should be larger than 1. The optimal value of *N* depends on the amount of available data and has to be determined through experiments. A general guide is that a small *N* may lead to a higher frame level performance but the sentence level performance may be lower. If *N* is too large, the number of misclassified vector-groups in the training data will be very small, in the extreme case, no wrong vector-groups, therefore, no learning happens. Another consideration is

how to select vectors to compose a vector-group. One approach is selecting N vectors continuously from the training sentences so that the dynamic information of speech signals is retained. Alternatively, the N vectors in a vector-group can be chosen randomly from the available training sentences. In this case, the vectors in one vector-group may come from different training sentences. In the text-independent speaker recognition, the spoken texts occurring in the training data may not exist in the test sentences, and the system relies mainly on the distributions of the feature vectors rather than on the order of vectors, thus it is not necessary to keep the order of the training vector sequences. In the following, all evaluation experiments were conducted in the text-independent mode, thus we make use of the second method to select vectors for each vector-group.

3. EVALUATION DATA

The proposed GVQ algorithm has been evaluated with the TIMIT database. Since the TIMIT database contains wideband (8 kHz) speech signals and were recorded in quiet environment, the identification task with this database is very easy. To demonstrate the power of the new training algorithm, we made the identification task more difficult by filtering all speech signals with a 101-point FIR filter. The bandwidth of the filter is 300- 3200 Hz (corresponding to the telephone line bandwidth). The GVQ algorithm is a discriminative training procedure, which means that data from all speakers should be presented during the training process. To reduce the total amount of required memory and experimental time, only a subset of the TIMIT database consisting of 112 male speakers were used. Besides, the unvoiced parts of speech signals are removed automatically based on an adaptive energy threshold. Seven sentences (two "sa" and five "sx" sentences) were used for training, and the rest three "si" sentences were used for test. The three "si" sentences were first concatenated together to form a long one and then was cut into several pieces of the same length (60 frames, 960 ms in length). 16 MFCC coefficients were calculated from each frame of the signals to compose a feature vector. The analysis window size was 32 ms (512 samples) with 16 ms overlapping. Only five test utterances from each speaker were used in the evaluation experiments, that is, on each test point, 112×5 classification decisions have been made.

4. EXPERIMENTAL RESULTS

4.1 Comparing the LBG and the LVQ algorithms

We first compare the performance of codebooks trained by the LBG or the LVQ algorithms. As mentioned before, the optimization criteria of these two algorithms are different. Figure 1 (a) shows the frame level classification rate (i.e., on the basis of individual test vectors) with these two kinds of the codebooks. The initial LVQ codebooks were taken from the LBG codebooks. It is seen that in all cases the classification performance was improved after the codebooks had been further fine-tuned by the discriminative LVQ algorithm. For LVQ codebooks, the classification rate degrades slightly in the case of using larger codebooks. This is because the LVQ codebooks have strong discriminative power and may get overfitted with the training data if the codebook is too large. On the other hand, the classification performance with the LBG codebooks improves monotonically with the codebook size. However, even at its highest point, it is still lower than the corresponding LVQ codebooks.

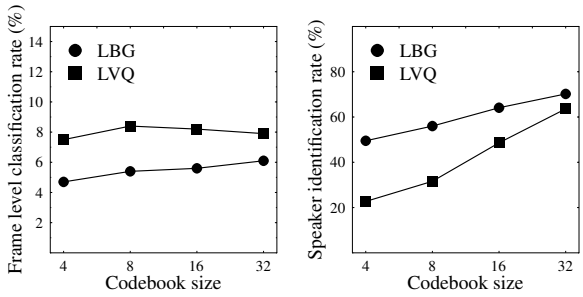

Figure 1 (a) Left panel: classification rate for individual test vectors; (b) Right panel: classification rate for test utterances. Each test utterance consists of 60 test vectors.

The sentence level performance is shown in the right panel of Figure 1. In contrast to the frame level situation, the LBG codebooks provide better sentence level performance than the corresponding LVQ codebooks. From the results shown in Figure 1, we see that, even though the LVQ codebooks give a higher classification rate for feature vectors, the speaker identification performance is not improved by applying the LVQ algorithm. The main reason is that the classification decision for a speaker depends on a vector sequence rather than on an individual vector. During the LVQ training procedure, only a single vector is considered each time and the correlation between these vectors is not taken into consideration. We will show that the proposed GVQ algorithm overcomes this weakness.

4.2 Parameters in the GVQ algorithm

Several parameters have to be specified before using the GVQ algorithm. The first one is the learning rate α. α determines how far the code vectors will be moved when a vector-group is misclassified. Other parameters include

the number of training epochs (ℓ) and the number of vectors in a vector-group (N). A training epoch is defined as the iteration number that equals to the total number of training vectors. We systematically studied these parameters with regard to their effects on the performance. Figure 2 displays the speaker identification rate as a function of the training epochs under different learning rates. For comparison, the performance from the initial LBG codebooks is also included in the figure. It is seen that the performance can always be improved after applying the GVQ algorithm.

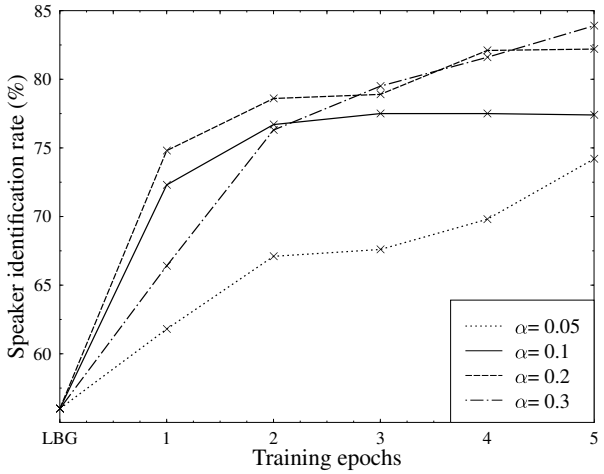

Figure 2 Learning rate (α) vs. learning epochs. The number of vectors in each vector-group (N) is 4, and the codebook size is 8.

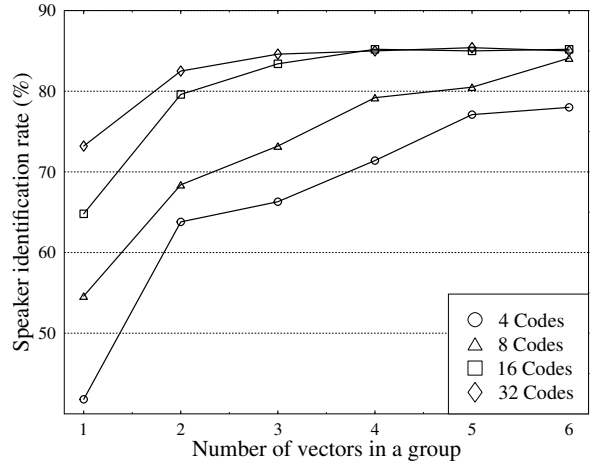

Figure 3 Speaker identification rate vs. codebook size and the number of vectors in each vector-group (N).

Now with α=0.2, ℓ=3, the effect of N is presented in Figure 3. Except for N=1, the GVQ codebooks can significantly outperform the corresponding LBG codebooks. With the increase of N, the performance improves further but it may need longer time to train the codebooks. The gain of using the GVQ algorithm is especially large for the smaller codebooks.

4.3 Comparing the GVQ codebooks with the GMM

A Gaussian mixture model (GMM) is a weighted sum of several multivariate Gaussian densities. Reynolds [4] first applied the GMMs to speaker recognition applications. Recently, the GMM becomes a popular speaker model and has been shown to give a very high speaker recognition performance. We also implemented this model and made a comparison with the VQ codebooks trained by different methods. A summary of the identification results is given in Table 1. The model order means the number of code vectors in the VQ codebooks or the number of Gaussian functions in the GMMs. As expected, the performance improves with the model order. For the same model order, the GMMs do provide a better performance than the LBG codebooks, but the GVQ codebooks give the highest performance. Another advantage of the VQ-based speaker models over the GMMs is that in the test phase the VQ models are faster.

Filtered TIMIT database				
Model Order	4	8	16	32
LBG	49.5	56.0	64.1	70.2
GMM	68.2	75.2	78.4	79.6
GVQ	74.3	79.8	85.9	86.5

Table 1 Comparing speaker identification performance with the LBG codebooks, the Gaussian mixture speaker models and the GVQ codebooks.

5. REFERENCES

[1] A. E. Rosenberg, and F. K. Soong, "Evaluation of a vector quantization talker recognition system in text independent and text dependent modes," Computer Speech and Language, Vol. 22, pp. 143-157, 1987.

[2] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Comm., Vol. 20, pp. 84-95, Jan. 1980.

[3] T. Kohonen, "The self-organizing map," Proc. IEEE, Vol. 78, pp. 1464-1480, 1990.

[4] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," Speech Communication, Vol. 17, pp. 91-108, 1995.