AN IMPROVED VQ BASED ALGORITHM FOR RECOGNIZING SPEAKER-INDEPENDENT ISOLATED WORDS

DING-DING MA, XIAO-QIN ZENG

Institute of Intelligence Science and Technology, Hohai University, Nanjing, China
E-MAIL: mdd_133@163.com, xzeng@hhu.edu.cn

Abstract:
In this paper, an improved codebook generation algorithm called SLVQ (Speaker Level Vector quantization) is proposed, which can improve the recognition accuracy of speaker independent isolated words. Linde-Buzo-Gary (LBG) algorithm is the most commonly used codebook design method. The idea behind LBG is to find an optimal codebook that minimizes the distortion between the training words and the codebook. But this does not guarantee that the testing words also have minimum distortion as training words. To address the problem of producing poor codebook for testing words in speaker independent speech recognition, the proposed method makes use of the diversity of different speakers by randomly selecting some speakers and their pronounced words in the codebook design procedure to optimize codebooks. An evaluation experiment has been conducted to compare the speech recognition performance of the codebooks produced by the LBG, the LVQ (learning vector quantization), and the SLVQ. It is clearly shown that the SLVQ method performs better than the other two methods.

Keywords:
Vector quantization; Codebook generation; Speech recognition; Speaker Independent

1. Introduction

Vector quantization (VQ) techniques have been successfully applied to solve the problem of isolated word recognition [1-3]. In general, a VQ-based speech recognizer is modeled by a number of codebooks derived from each class of training words. Recognition involves selecting the codebook which quantizes an unknown word with minimum distortion. Thus, the codebook is the heart of a VQ-based speech recognizer.

At present, the most popular codebook design algorithm is LBG [4]. The LBG iteratively performs two steps, ie. the partition step and new codebook generation step, based on the criterion of minimizing the quantization error until convergence. Unfortunately, since the distortion function is not convex and may contain multiple local minima [5], the LBG often produces nonoptimal codebooks. These nonoptimal codebooks cause a poor performance in speech recognition or speaker identification. A number of methods have been proposed in the literature to overcome this nonoptimal codebooks problem. They include global optimization methods [6-7], discriminative training methods [8-10]. In order to improve the speaker identification performance, the matrix quantization (MQ) method is proposed in [11]. The MQ method generates codebooks using a block of vectors instead of a single vector in the feature matrix. But both VQ and MQ are not suitable for speaker independent speech recognition because the codebook of a word is trained solely from the training words of that speaker. In speaker independent speech recognition, the classification decision is made by all speaker information of each vocabulary model. So, minimizing the classification error rate is a more suitable optimization criterion for speaker independent speech recognition.

LVQ [12] is a discriminative training algorithm proposed by Kohonen. LVQ training is to reduce the number of misclassified vectors, which does not take into account the fact that speech feature vectors are highly correlated. When LVQ training is applied to speaker identification the correct classification rate may be degraded. In order to overcome this weakness, a modified version of the LVQ is proposed in [9]. However, the [9]’s method is not suitable to speaker-independent isolated word recognition. The lack of effective methods to improve VQ-based speaker independent isolated word recognition accuracy motivates us to search for a solution to the problem, and this result in the proportion of SLVQ algorithm.

The main contribution of the paper is that it proposes an improved VQ based algorithm, which could highly improve the speaker-independent isolated word recognition accuracy, and demonstrates effectiveness of the proposed algorithm SLVQ by experiments in comparison with the LBG and LVQ algorithms.

The remainder of the paper is organized as follows. Section 2 gives a brief description to the LBG and LVQ algorithms and points out the difference between the two
training algorithms. The SLVQ training algorithm is presented in Section 3. The recognition experiments carried out by LBG, LVQ and SLVQ and their results with comparison are given in Section 4. Section 5 concludes the paper.

2. Vector quantization algorithms

This section briefly describes two popular algorithms for generating codebooks, and discusses their difference and weakness for applying to speaker independent isolated word recognition.

2.1. LBG algorithm

The SLVQ algorithm described in this paper is based on the LBG algorithm [4]. This algorithm starts by assigning a long training sequence that is chosen to be representative of all speech data to a single cluster and proceeds by binary splitting until the desired number of clusters is achieved. In this paper, we designed each codebook from a training sequence containing repetitions of one word spoken by different people in the recognition vocabulary. In the recognition phase, the unknown word is classified according to the codebook yielding the lowest average distortion. The description of the LBG algorithm is as follows:

Step 1. Set \( m=1 \). Calculate centroid \( C_1 = \frac{1}{T} \sum_{j=1}^{T} X_j \), where \( T \) is the total number of training data vectors. The set of \( k \)-dimensional training data vectors was denoted by \( X = \{ X_j \mid X_j \in \mathbb{R}^k; j=1,\ldots,T \} \), the set of codewords was denoted by \( C = \{ C_i \mid C_i \in \mathbb{R}^k; i=1,\ldots,N \} \).

Step 2. Divide each centroid \( C_i \) into two close vectors \( C_{2i-1} = C_i * (1 + \delta) \) and \( C_{2i} = C_i * (1 - \delta) \), \( 1 \leq i \leq m \). Here, \( \delta \) is a small fixed perturbation scalar. Let \( m=2m \). Set \( n=0 \), here \( n \) is the iterative times.

Step 3. Find the nearest neighbor to each data vector. Put \( X_j \) in the partitioned set \( P_i \) if \( C_i \) is the nearest neighbor to \( X_j \).

Step 4. After obtaining the partitioned sets \( P = \{ P_i \mid 1 \leq i \leq m \} \), set \( n=n+1 \). Calculate the overall average distortion \( D_n = \frac{1}{T} \sum_{i=1}^{n} \sum_{j=1}^{r_i} D(X_j^{(i)}, C_i) \), where \( P_i = \{ X_1^{(i)}, X_2^{(i)}, \ldots, X_{r_i}^{(i)} \} \).

Step 5. Find centroids of all disjoint partitioned sets \( P_i \) by \( C_i = \frac{1}{r_i} \sum_{j=1}^{r_i} X_j^{(i)} \).

Step 6. If \( (D_{n-1} - D_n) / D_n > \varepsilon \), go to step 3; otherwise go to step 7. Here, \( \varepsilon \) is a small distortion threshold.

Step 7. If \( m=N \), then take the codebook \( C_i \) as the final codebook; otherwise, go to step 2. Here, \( N \) is the codebook size.

2.2. LVQ algorithm

LVQ algorithm is a well-known discriminative training procedure in which the codebooks of all classes are trained together and the code vectors are modified depending on the local difference of density functions [12]. The learning rule of LVQ is as follows [10]:

First, we calculate Euclidean distance for all code vectors \( C_i(t) \) and then find the closest code vector \( C_m(t) \) to the input vector \( X(t) \).

\[
\| X(t) - C_m(t) \| = \min \{ \| X(t) - C_i(t) \| \} \tag{1}
\]

Let the input vector \( X(t) \) belong to category \( S_r \) and the closest code vector \( C_m(t) \) to \( X(t) \) belong to \( S_s \). If \( S_r \) belongs to the same category as \( S_s \), \( C_m(t) \) is moved closer to \( X(t) \) as shown in Eq.(2). If \( S_r \) belongs to a category other then \( S_s \), \( C_m(t) \) is moved away as shown in Eq.(3). All code vectors except \( C_m(t) \) are not updated. The moving direction is \( X(t) - C_m(t) \).

\[
C_m(t+1) = C_m(t) + \alpha(t) \{ X(t) - C_m(t) \} \tag{2}
\]

\[
C_m(t+1) = C_m(t) - \alpha(t) \{ X(t) - C_m(t) \} \tag{3}
\]

Where the function \( \alpha(t) \) ought to be a monotonically decreasing function of time \( 0 < \alpha(t) < 1 \). The code vectors are initialized by those sampled randomly from all the input vectors. Then the code vectors are adjusted by moving closer or away from the input vector based on Eq. (2) and Eq. (3).

The codebook trained by LBG algorithm is in the sense that the quantization error is minimized. The criterion is more suitable for the application of data compression than speech recognition. Because speech recognition is in the pursuit of high recognition performance, but minimum quantization distortion does not mean the best recognition performance. So, the proper criterion for training codebook is to minimize the classification error rate. LVQ is a discriminative training algorithm that can improve the classification correctness rate for a single vector. But LVQ could hardly improve the performance of speaker independent isolated word recognition, and in some cases the word recognition accuracy may even be degraded.
3. **SLVQ training algorithm**

In this section, we present in detail the SLVQ training algorithm and point out where and how it differs from the LVQ algorithm and the method proposed in paper [9].

As mentioned in section 2, the recognition of speaker independent isolated word is determined by the codebook derived from a training sequence of several repetitions of a word spoken by several persons rather than by a single person. The codebook trained by LVQ can improve the speech vector classification rate, but the word recognition accuracy can not be improved. In order to improve the performance of word recognition, we propose the following training method, the SLVQ, to optimize the codebooks for speaker independent isolated word recognition. Suppose the training set include some vocabularies containing repetitions of each word spoken by different persons. The training algorithm of SLVQ is given as follows:

**Step 1.** Use the LBG algorithm to train and generate the initial codebook for the training set. The number of codebooks is the same as the number of words.

**Step 2.** Randomly select a speaker, then randomly take a word from the training words spoken by that speaker, and use label A to denote the selected word.

**Step 3.** Calculate average quantization distortion $D(i)$ using Eq. (4) for all word codebooks (indicates the word codebooks’ label). If the codebook of word B gives the minimum quantization distortion, go to step 4, otherwise go to step 5.

\[
D(i) = \frac{1}{N} \sum_{i=1}^{N} \min \| X_i - C_j \|, 1 \leq j \leq M \quad (4)
\]

Here, $N$ is the number of speech frame of a word, and $M$ is the codebook size.

**Step 4.** If $(D(A) - D(B))/D(A) < \delta$, where $\delta$ is a preselected small distortion threshold. For each frame vector $X$, find its nearest code vector from the codebook of word A (denoted as $C(A)$), and the nearest code vector from codebook of word B (denoted as $C(B)$). Adjust the two code vectors simultaneously by Eq.(5) and Eq.(6). If the ratio $(D(A) - D(B))/D(A)$ is larger than the value $\delta$, none of the code vectors need to be adjusted. After finishing this step, go to step 6.

\[
C_m^{(A)} = C_m^{(A)} + \alpha (X_i - C_m^{(A)}) \quad (5)
\]

\[
C_m^{(B)} = C_m^{(B)} + \alpha (X_i - C_m^{(B)}) \quad (6)
\]

Where $\alpha$ is the learning rate, its value decreases with the number of iterations increase. Here, we set $\alpha = \alpha * \text{loops} / \text{total_loops}$ as the initial value.

**Step 5.** In this case, the selected word A is correctly recognized. We only adjust the code vectors of word A’s codebook. For each frame vector $X$, in the word A, adjust its nearest code vector in the codebook of word A by Eq. (7). After finishing this step, go to step 6.

\[
C_m^{(A)} = C_m^{(A)} + \beta \alpha (X_i - C_m^{(A)}) \quad (7)
\]

Here, $\beta$ is momentum coefficient.

**Step 6.** Go to step 2 until the iteration number is larger than the desired number, otherwise the training procedure stops.

It’s worth mentioning that SLVQ differs from LVQ mainly in the criterion used for adjusting the code vectors. In the LVQ case, the criterion is to improve the correct vector classification rate for all words, while in the SLVQ case, the criterion is to improve the correct word classification rate for all speakers. The difference between SLVQ and the method proposed in [9] is that SLVQ could distinguish each word spoken by a specified speaker and [9]'s method could not distinguish each single word because it puts all words spoken by the speaker together in the training. Thus, [9]'s method is more suitable for speaker identification than word recognition.

4. **Recognition experiment and results**

To evaluate the recognition performance of the proposed algorithm, LBG, LVQ and SLVQ algorithms were coded in C language and run in Matlab environment through MEX-files. In this section, we first describe the database and the experimental setups used in the experiments, and then describe the experimental results with a comparison among the three methods.

4.1. Database and experimental setup

The database was taken from the Arabic digit corpus collected by laboratory of automatic and signals at the University of Badji-Mokhtar in Annaba, Algeria. This database was created from all ten Arabic digits. A number of 88 individual (44 males and 44 females) Arabic native speakers were asked to utter all digits ten times. Depending on this, the database consists 8800 utterances (10digits×10repetitions×88speakers). In this experiment, the dataset is divided into two parts: a training set with 6600 utterances (66 speakers containing 33 males and 33 females) and test set with 2200 utterances (22 speakers containing 11 males and 11 females). In this research, speaker independent mode is considered [13].

In this experiment, the speech signal were
pre-emphasized by the filter $H(z) = 1 - 0.97 z^{-1}$.

Hamming window is applied to the pre-emphasized signal of a given frame. This speech frame was represented by 13 Mel Frequency Cepstral Coefficients (MFCCs). In the training procedure by means of SLVQ, we set the threshold $\delta = \text{loops} / \text{total_loops} + 0.1$, the momentum coefficient $\beta = 0.1$.

4.2. Results from LBG, LVQ and SLVQ

We evaluate the recognition performance of the SLVQ training method by running experiments on the data in the Arabic digit database and compare the results with that of the LBG and LVQ methods. The word recognition accuracy results are listed in Table 1.

<table>
<thead>
<tr>
<th>Codebook size</th>
<th>Methods</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>LBG</td>
<td>84.50</td>
<td>91.23</td>
<td>92.32</td>
<td>94.41</td>
<td>94.45</td>
<td>95.09</td>
<td>92.00</td>
</tr>
<tr>
<td>8</td>
<td>LVQ</td>
<td>10.00</td>
<td>59.64</td>
<td>41.41</td>
<td>91.00</td>
<td>94.45</td>
<td>94.09</td>
<td>65.10</td>
</tr>
<tr>
<td>16</td>
<td>SLVQ</td>
<td>85.73</td>
<td>91.82</td>
<td>95.05</td>
<td>96.95</td>
<td>98.32</td>
<td>97.41</td>
<td>94.21</td>
</tr>
</tbody>
</table>

In order to make a visual comparison among the three methods, we plot the word recognition accuracies as a function of codebook size in Fig.1. We can make the following two observations from Fig.1: (1) the three methods reach the highest performance when codebook size is 64; (2) the SLVQ method provides the best recognition results among the three methods. This shows the effectiveness of the SLVQ method for improving the performance of speaker independent isolated word recognition.

5. Conclusions

In this paper, we have proposed the SLVQ algorithm to generate codebook for improving speaker-independent isolated word recognition accuracy. The method iteratively executes two steps, namely random selections of a speaker and a word spoken by the speaker, during training for optimizing the codebook initialized by LBG and capturing the dynamic difference of various speakers. The speech recognition performance of the SLVQ is evaluated by experiments on the Arabic digit database and the experimental results show that SLVQ's recognition performance is better than both the LBG and LVQ's.

Acknowledgements

This work is partially supported by the National Natural Science foundation of China under grant 60971088.

References


