A FAST TWO-LEVEL SPEAKER IDENTIFICATION METHOD EMPLOYING SPARSE REPRESENTATION AND GMM-BASED METHODS

H. Zeinali(1), H. Sameti(1), H. Khaki(2), B. Babaali(1)

(1) Department of Computer Engineering, (2) Department of Electrical Engineering
Sharif University of Technology
zeinali@ce.sharif.edu, sameti@sharif.edu, khaki@ee.sharif.edu, babaali@ce.sharif.edu

ABSTRACT

In large population Speaker Identification (SI), computation time has become one of the most important issues in recent real time systems. Test computation time depends on the cost of likelihood computation between test features and registered speaker models. For real time application of speaker identification, system must identify an unknown speaker quickly. Hence the conventional SI methods cannot be used. In this paper, we propose a two-level method that utilizes two different identification methods. In the first level we use Sparse Representation to decrease the search space. In the second level we use GMM-based SI methods to specify the target speaker. We achieved 18× speed-ups without any loss of accuracy using the proposed method.

1. INTRODUCTION

As the security concerns increase, the necessity for using identification systems grows up. In this area there are several unique features which can be used to distinguish the identity of a person e.g. fingerprint, iris, face and speech. An identification system can be used in practice only if it can work in real time. So this is an essential issue for any identification system.

The goal in Speaker Identification (SI) task is to distinguish the producer of an unknown input voice utterance. Conventionally, Gaussian Mixture Model (GMM) is used to model speaker features in training stage and in the test stage the speaker whose model has the maximum probability of test features, is selected [1]. In large population applications of SI, likelihood computation between unknown input voice and speaker models is very time consuming. For real time application we cannot use conventional GMM-based methods because these methods are very slow during test stage, although these methods have good accuracy in large population SI [1].

There are several different techniques to reduce the identification time (computation time) in the test stage. Vijendra et al. [2] proposed a GMM-based speaker model clustering method. In this method the speaker models are clustered using k-means algorithm in training stage. First, the best clusters are selected during the test stage and then the search space will be just between speaker models in these clusters. They show that by this method, 4.4× speed-ups are achieved with little to no loss in identification accuracy.

Sarkar et al. [3] proposed a Maximum Likelihood Linear Regression (MLLR) based method to calculate likelihood from the speaker models more quickly. In this method there are two-levels. In the first level they quickly find the best N speakers by using MLLR-based fast calculating method, during the test stage and in the second, they select target speaker out of N best speakers by using a conventional SI method. They show that this method performs faster than GMM-UBM based system [4] with some loss in identification accuracy.

Reynolds et al. [4] propose a GMM-UBM method based on top C-best mixtures for likelihood computation. In the test stage to find the best scoring mixture components, test utterance is first scored against UBM and then indices of the dominant Gaussian components for every frame are used for likelihood computation. This method can do the SI task about 5 times faster but there is a tradeoff between the accuracy and speed-up that is indicated in the number of selected best mixtures (C-best).

There are several other techniques which try to solve the slowness in speaker identification in test stage. Pre-Quantization (PQ) technique proposed in [5] tries to select a subset of the test utterance features vector instead of the whole data. This subset can be selected randomly and also it can be the average of different frames features or the representative of a clustering method on the feature vector. Another technique is speaker pruning [6] which is a step by step method to identify the speaker. In each step, a small portion of the test utterance is used to prune the speaker set and at the last step a unique speaker is resulted.

In this paper we focus on speed in test stage for large population Speaker Identification. It is known that in SI systems accuracy is very important, hence speed up must not effect on SI accuracy. Sparse representation [9] as the state of the art method in speaker identification has been used. By selecting N-best speakers out of the sparse method result, faster method than conventional methods is resulted.
The rest of this paper is organized as follow. In Section 2, the basic SI methods are explained. Our proposed method which is a combination of sparse technique and conventional GMM-based is described in Section 3. Experiments and results on TIMIT corpus is given in Section 4. The experimental results and the discussions are presented in Section 5. The paper is concluded by the Section 6.

2. IMPLEMENTATION METHODS

2.1. Baseline system

In this paper we implement GMM-based Speaker Identification method as baseline system. In this method each speaker’s utterance features are modeled using a single GMM in the training stage. A GMM is a weighted sum of M component of a multidimensional Gaussian density function as given by the below equation

\[ p(X | \lambda) = \sum_{i=1}^{M} w_i N(x | \lambda_i, \Sigma_i) \]

where \( x \) is a D-dimensional continuous-valued data vector, \( w_i, i = 1, \ldots, M \), are the mixture scalar weights, and \( N(x | \lambda_i, \Sigma_i), i = 1, \ldots, M \), are the component Gaussian densities. The mixture weights must satisfy the constraint that \( \sum_{i=1}^{M} w_i = 1 \).

For a sequence of \( T \) vectors \( X = \{x_1, \ldots, x_T\} \), the GMM likelihood with independence assumption between the vectors, can be written as

\[ p(X | \lambda) = \prod_{i=1}^{T} p(x_i | \lambda). \]

And in log domain

\[ \log(p(X | \lambda)) = \sum_{i=1}^{T} \log(p(x_i | \lambda)). \]

In the testing stage, \( T \) feature vectors \( x_{i\text{test}}, i = 1, \ldots, T \) are extracted out of a test utterance. Then the probability is calculated for all \( S \) speaker models using a log-likelihood calculation as in Equation (3) and the most likely speaker identity \( s \) decided according to

\[ s = \arg \max_{1 \leq s \leq S} \sum_{i=1}^{T} \log(p(x_{i\text{test}} | \lambda_i)). \]

For more details refer to [1].

2.2. Speaker Model Clustering Method

In GMM-based method we must calculate test features probability for all registered speakers as in Equation (3). This computation is very time consuming. In Speaker Model Clustering method [2] first we cluster all speakers in \( N \) clusters in the training stage. Then in the test stage we just search over those clusters containing the most probable speakers. By this method we can reduce the search space. In this paper we just implement the best clustering methods proposed in [2] for compression. This method is called Kullback–Leibler, GMM-Based Clustering.

2.3. Sparse Representation

First assume that we have \( S \) speakers and have sufficient training utterances \( n_i \) for the \( i \)th speaker. Each training utterance has a different length that needs to be mapped on a fixed-dimension feature vector space using the GMM mean supervector [7]. In this mapping first we train a single GMM-UBM for each utterance. Then we append GMM-UBM mean vectors to each one and make a large vector called the mean supervector. We named the new feature vector as \( v_{i,j} \) that \( i \) is the index of the speakers, \( 1 \leq i \leq S \) and \( j \) is the index of the training utterance, \( 1 \leq j \leq n_i \). Then we create matrix \( A_i = [v_{1,1}, v_{1,2}, \ldots, v_{i,n_i}] \) for each speaker (dictionary matrix for speaker \( i \)). Let \( y \) be the GMM-UBM mean supervector for a test utterance from the \( i \)th speaker. Then \( y \) will approximately lay on the linear span of the training utterance associated with speaker \( i \)

\[ y = \alpha_{i,1} y_{i,1} + \alpha_{i,2} y_{i,2} + \ldots + \alpha_{i,n_i} y_{i,n_i}. \]

After that we create a dictionary matrix \( A \) for all \( S \) speakers as follow:

\[ A = [A_1, A_2, \ldots, A_k]. \]

The test vector \( y \) can now represent in terms of all training samples as follow:

\[ y = Ax, \]

where

\[ x = [0, \ldots, 0, \alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i}, 0, \ldots, 0]^T. \]

If we solve Equation (6) to find \( x \) we actually find the class of the test vector \( y \). We are able to solve this problem using the \( l_1 \)-norm minimization [8]:

\[ (l_1^i): \hat{x}_i = \arg \min_{x} \|x\|_1 \text{ subject to } \|Ax - y\|_2 \leq \varepsilon. \]

For test vector \( y \) we first compute its sparse representation \( \hat{x}_i \) from (7). Ideally \( \hat{x}_i \) must have nonzero entries associated with columns of \( A \) from a single speaker \( i \). Due to the noise and other limitations there are some small entry on columns associated to other speakers. To solve this problem we defined a function \( \delta_i \) for each speaker \( i \) that \( \delta_i(\hat{x}_i) \) return a vector whose only nonzero entries are the entries of \( \hat{x}_i \) that are associated to speaker \( i \).

At the end according to the (8) we find the target speaker:

\[ s = \arg \min_{1 \leq s \leq S} \|y - A\delta_i(\hat{x}_i)\|_2. \]

For more details on sparse representation refer to [9], [10].

3. PROPOSED METHOD

In large population SI it seems if we have a fast method without excellent accuracy, we can use two-level methods to compensate the accuracy loss. In testing sparse representation for Speaker Identification [9], this method was shown to be very fast for large population SI task, hence we can use it in the first stage of our method.
In this paper we implement Sparse Method as explained in [9]. First we trained a UBM from all speaker training utterances. Then we adapt a GMM-UBM out of UBM model for each speaker utterance. Finally, we applied this method on 630-speaker TIMIT. This method has identification accuracy of 96.34% alone. In each test we sorted speakers based on Sparse Identification results and we found that in most cases target speaker is in $n$ first speakers. Therefore we selected the $n$ best speakers from the results generated by this method. We selected the target speaker from the $n$ best using conventional GMM-based methods at second level. By this two-level method we achieved significant speed-up in average without any loss in accuracy. Actually we increased SI accuracy of sparse method and also we presented a fast method that can be used for real time applications.

There are several methods for sparse coding. In this paper we use the fast implementation proposed as Efficient Sparse Coding Algorithms [8]. This sparse coding method can solve Equation (7) rapidly.

4. EXPERIMENTS AND RESULTS

We implemented Speaker Model Clustering method [2] to be compared with the speed-up performance of our proposed method. Also we implemented conventional GMM-based method as the baseline [1]. First we used an energy based voice activity detector to remove silence from utterances. In this paper we used 630-speakers TIMIT corpus [11]. In TIMIT each speaker has 10 utterances containing a single sentence. In all tests we used two dialect sentence (SA) utterances as the test speech, in one case we used both sentences together (Two-Utterance) and for the other one they were separately tested (One-Utterance). The other 8 sentences were used for training. For GMM-based method as our baseline method, we used feature vectors comprising 29 MFCCs including $C_0$. We modeled every speaker by a 32-mixture GMM. We used HTK [12] for training speaker models in all experiments. The trained speaker models were also used for speaker model clustering.

In our approach for the first level we performed as in [9]. In this method we used 13 MFCCs including $C_0$ feature vectors. First a Universal Background Model (UBM) was trained with all training utterances of all speakers. UBM had 64 mixtures that were trained by the HTK. In the next step, we used map-adaptation to update UBM mixtures means. In this stage we trained a single model per utterance. In other words each speaker had 8 training models. In the first level we used sparse representation speaker identification of [9]. For the second level, we trained a 32-mixture GMM for each speaker out of the speaker’s training utterances exactly as in the GMM-based method. In the test stage we selected $n$ best speakers out of all registered speakers that got the best results in sparse identification stage. Then, we did likelihood computation only for the $n$ selected speakers and the target speaker was selected.

In the First test, the goal was to find a proper value for $n$ to have a reasonable accuracy. After doing sparse identification $n$ best speakers were selected. We used the existence rate of the target speaker over the $n$ best selected speaker as a criterion for first level accuracy. In Fig. 1 first level accuracy as a function of number of selected speakers has been shown.

![Figure 1. First level accuracy as a function of number of selected speakers](image)

<table>
<thead>
<tr>
<th>Testing Method</th>
<th>Speedup</th>
<th>Identification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-base</td>
<td>1×</td>
<td>99.12%</td>
</tr>
<tr>
<td>Clustering</td>
<td>2.8×</td>
<td>97.38%</td>
</tr>
<tr>
<td>Two-Level</td>
<td>11.3×</td>
<td>99.29%</td>
</tr>
</tbody>
</table>

For speaker model clustering method we just reported the result where the search space was over 20% of all clusters. Our clustering implementation results were not the same as the results reported in [2] and we will explain the reason in discussion Section. As you can see our method achieved 11.3× speed-ups and also increased the identification accuracy.
At the second experiment we used 2 utterances as one identifying test (Two-Utterance). In this case the average length of test speech was about 6 seconds per speaker. For GMM-based and speaker clustering we performed similarly to the One-Utterance identifying test. For sparse representation first we scored each utterance separately and then merged the two results and 10-best speakers were selected out of the merged results. In Table II the outcomes of this experiment are reported.

Table II. Average speed-up factors and SI accuracy for Two-Utterance test on TIMIT corpus.

<table>
<thead>
<tr>
<th>Testing Method</th>
<th>Speedup</th>
<th>Identification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-base</td>
<td>1×</td>
<td>100%</td>
</tr>
<tr>
<td>Clustering</td>
<td>2.8×</td>
<td>99.84%</td>
</tr>
<tr>
<td>Two-Level</td>
<td>18×</td>
<td>100%</td>
</tr>
</tbody>
</table>

It can be seen that the two-level method results in proper accuracy with significant speed-up.

5. DISCUSSION

The main goal of this work was to speed-up Speaker Identification task. As it was shown by two-level method and employing the sparse technique we got not only a fast system about more than 18 times but also without any loss in identification accuracy. This can be compared to the other state of the art methods which reported about 4.4 times faster at best [2].

By comparing Table I and Table II it is clear that by increasing test speech length, speed-up of our method increases.

The logic behind of our work can be interpreted as follows. Experiment shows that calculation time of sparse representative for 630 numbers of speakers is equal to (630/(2*12+10)) times faster than the base method. In [2], they first calculate 200 probabilities and at most it will get 3.15 (≈630/(100+120-20)) times faster than the base method. Hence totally they need to calculate 100 likelihoods between cluster representative models and then by searching over the 20-best clusters which in average consist of 120 (=630/5) speakers, the target speaker is found (20 probabilities calculated during first stage). Hence totally they need to calculate 200 probabilities and at most it will get 3.15 (≈630/(100+120-20)) times faster than the base method.

6. CONCLUSIONS AND FUTURE RESEARCH

In this study, sparse technique helped to find a sparse representation quickly and with selecting the n-best speakers, we reduced the search space. The result of this paper indicates that the speed of the speaker identification task is increased up to 18 times.

Although in this paper we designed an 18 times faster test stage, but the identification time is still high and it will increase as the number of speakers increases. A future study could investigate to find other fast techniques that can be substituted for sparse level.

7. ACKNOWLEDGMENT

This research was partially supported by ITRC (Iran Telecommunication Research Center).

REFERENCES