Speaker Detection using Acoustic Event Sequences

Nicolas SCHEFFER, Jean-François BONASTRE

LIA-CNRS, Avignon (France)
{nicholas.scheffer, jean-francois.bonastre}@lia.univ-avignon.fr

Abstract

Novel approaches using high level features have recently shown up in the speaker recognition field. They basically consist in modeling speakers using linguistic features such as words, phonemes, idiolects. The benefit of these features was demonstrated in NIST campaigns. Their main disadvantage is their need of a high amount of data to be efficient. The purpose of this study is to generalize this approach by using acoustic events, generated by a GMM, as input features. A methodology to build a dictionary and to model speakers using symbol sequences from this dictionary is derived. Different experiments on NIST SRE 2004 database show that the information produced is speaker specific and that a fusion experiment with a GMM verification system improves performance.

1. Introduction

In the past few years, the automatic speaker recognition field made an extensive use of Gaussian Mixture Models to answer the problem of text-independent speaker recognition [1]. The latter has fulfilled its role by reducing error rates especially in NIST SRE campaigns\(^1\). However, since 1998, global performance follows an asymptotic curve, and a limit seems to have been reached. Since the introduction of an extended data task, drastically increasing the amount of data, the community tends to model other types of information, prosodic, phonetics \cite{2}, idiolectical \cite{3}, usually referred as high level features \cite{4}. The first step in building the dictionary consists in extracting phoneme sequences for a dictionary used by the LIA for the NIST SRE 2004 campaign (female only). The training is performed based upon the NIST 2004 database, primary condition, all trial set, female speakers only. This condition consists of 370 speakers. Train and test utterances contain 2.5 minutes of speech in average (telephone conversation). The whole speaker detection experiment consists in 14717 tests (1320 target tests). Each test is made independently and the use of information from other tests to take a decision on the current test is forbidden.

3. Speaker independent acoustic dictionary

The dictionary symbols are generated by a speaker independent Gaussian Mixture Model, usually referred to as background model. The amount of training data is voluntarily high so that information contained in the GMM is maximal.

The first step in building the dictionary consists in extracting the phoneme sequences, hence the number of possible symbol sequences for a \(N\) order modeling approach can reach \(2048^N\). As these sequences are used to compute the detection test, elaborating a method to reduce the dictionary size seems necessary. After giving information on the background model used in this paper, a process to cluster Gaussian indexes into classes is presented.

3.1. Training a background model

The background model used for the experiments is the same as the background model used by the LIA for the NIST SRE 2004 campaign (female only). The training is performed based on NIST SRE 1999 and 2002 databases, and consists in 5 millions of speech frames (13.5 hours). Training was performed using the ALIZE and LIA_SpkDet toolkits\(^2\) \cite{5}. Frames are composed of 16 LFCC parameters and its derivatives. A normalization process is applied, so that the distribution of each cepstral coefficient is 0-mean and 1-variance. The background model posses 2048 components and no component variance is above 0.5.

---

\(^1\)www.nist.gov/speech

\(^2\)http://www.lia.univ-avignon.fr/heberges/ALIZE/
3.2. Confusion Matrix Computation
A function associating a symbol to a class containing multiple Gaussian indexes is presented in the following. The first step is to build a confusion matrix to analyze the symbols generated by the background model. It aims at giving indicators of the confusion degree between Gaussian so that they can be merged into higher level classes. The size of the matrix is $N \times N$, $N$ being the number of components in the GMM. The matrix is computed using the indexes of the GMM "top-ten" components. Let $M$ be the confusion matrix and $G_1, \ldots, G_N$ the top N Gaussian for a frame. Then, for each frame, the updating process of the matrix is given by $M(G_i, G_k)$ for $k \in [1, N]$. We end up with a matrix where on each row $i$:
- elements on the diagonal correspond to the number of times the Gaussian $i$ is in first position.
- other elements indicate the presence of the corresponding Gaussian in the top ten positions where $G_i$ was first.

3.3. Reducing the dictionary size
The next step consists in clustering the different symbols into classes by using a minimum confusion criterion. To perform this, confusion rates, noted $\tau$, are computed using the above matrix as $\tau = M(i, j) + M(j, i)$. Using these indicators, we proceed thus:
- Class merging is done sequentially by searching for $(C_i, C_j)$ where $\tau$ is maximum.
- In case of an equal confusion rate, priority is given to the smallest class hence avoiding an attractive class to be generated.
- The classes $(C_i, C_j)$ are then merged in a new class $C_i' = (C_i, C_j)$.

3.4. Pruning
In order to smooth out the symbol distribution and to get rid of non informative classes, an Out Of Vocabulary class, usually referred to as OOV class is created. A pruning threshold $N$ is also given to get rid of classes that have never been more than $N$ times at the top position.

3.5. Example
To illustrate this process in practice, when reducing from 2048 to 64 symbols and with a pruning threshold of 10, 573 OOV classes were removed in the first place. The resulting dictionary has the following properties: a mean of 23 Gaussian per class, a median of 17 and a standard deviation of 22. The dictionary building process is summarized in figure 1 part A.

3.6. Symbol generation procedure
The adopted strategy is to transform all the parameterized signals into symbol sequences. Indeed, each feature file belonging to the train, test and world model set is submitted to the same process. For a given utterance, each frame is passed through the background model to compute its 1-best component. Then, the index of this component is replaced with its corresponding symbol in the dictionary previously built. It is worth noting that the resulting sequence length is the same as the number of frames in the feature file. This procedure is summarized in figure 1B.

4. Speaker specific information modeling
Having produced a speaker independent dictionary, the following presents a method to build speaker specific models upon this dictionary. This aims at modeling speaker specific symbol sequences. Indeed, as well as for phonemes in [2] and idiolect in [3] where sequence analysis has been granted as speaker specific, the main idea of the study is to generalize this approach to non-linguistic features, such as the acoustic events defined by the dictionary symbols.

4.1. Acoustic symbol sequence modeling
This section describes the modeling method adopted to build speaker specific models based on a Ngram approach. The first paragraph presents Ngram models, next a technique to minimize unseen events is shown.

4.1.1. Building Ngram models
The Ngram modeling technique used in this paper is a "bag of Ngram" approach, following the work of [3]. It consists in transforming a sequence of symbols, generally in a temporal order (words, pitch dynamics, phonemes, . . .) into tokens $k$ with their associated probability $p(k)$.

4.1.2. Dealing with unseen events
In dealing with unseen events, the most known strategy is to use back-off techniques. It tends to revert to the (N-1)-gram probability to compute the score of the token. In our case, we chose the MAP adaptation solution given by [6] and expressed as follows:

$$\tilde{C}_M(k) = C_M(k) + \alpha C_{BM}(K)$$  \hspace{1cm} (1)

where the $C_M(k)$ is the count of the token $k$ in the model $M$ training data and $\alpha$ is an adaptation weight in the range (0,1). The estimated likelihood becomes:

$$\tilde{\ell}_M(k) = \frac{\tilde{C}_M(k)}{\sum_k \tilde{C}_M(k)}$$  \hspace{1cm} (2)

5. Detection Test computation
Two techniques in order to compute a log-likelihood ratio (LLR) in Ngram based modeling strategy are described. The classical likelihood ratio and a SVM based method, specifically designed to compute the LLR of Ngram based models. The whole speaker verification system is illustrated in figure 2.
5.1. Log-Likelihood Ratio of a test segment

The score of a test segment is estimated in terms of a likelihood ratio resulting from a test between a speaker model $S$ and a background model $BM$. The test segment score is then expressed as follows:

$$\ell \ell l_S = \frac{1}{C(k)} \cdot \sum_k C(k) \log \left( \frac{\ell_S(k)}{\ell_{BM}(k)} \right)$$

where $S, BM, C(k), \ell_S(k)$ and $\ell_{BM}(k)$ are the speaker model, the background model, the count of token $k$ in test, and their associated likelihoods for the token $k$ respectively.

5.2. TFLLR weighting using SVM

The TFLLR (Term Frequency LLR) method, based on SVM classifiers, has been proposed by Campbell in [7] and has proved to reduce error rates comparing to the standard LLR classifiers, has been proposed by Campbell in [7] and has proved to reduce error rates comparing to the standard LLR classifiers.

5.2.1. Kernel construction

To build a suitable kernel function, the likelihood ratio equation expressed with Ngram tokens (eq. 3) is used. Knowing that $\sum_k C(k)$ is constant, TFLLR method express the LLR as follows:

$$\ell \ell l_S = \sum_k \ell_I(k) \cdot \log \left( \frac{\ell_S(k)}{\ell_{BM}(k)} \right)$$

The token $k$ likelihood in the test is estimated by the likelihood of $k$ in the impostor model $I$. After a linearization process, the equation becomes:

$$\ell \ell l_S = \sum_k \frac{\ell_I(k)}{\ell_{BM}(k)} \ell_S(k) - 1$$

The kernel construction finally resides in the weighting of speaker likelihoods by the likelihood of the background model.

5.2.2. Maximum margin decision

In order to build impostor models, the background model data has been divided into 1000 separate subsets. The input of the classifier is the concatenation of all impostor trials and the target speaker trial. The maximum margin decision is found by passing this input through a linear kernel using the Torch toolkit[8]. The output of the test is used as a score for verification.

6. Experiments

All experiments were performed using the NIST SRE 2004 database, female subset. Tests were performed on the 1-side/1-side test condition. Results were compared in terms of equal error rates and DET curves.

6.1. Scoring with standard LLR

Figure 3(a) shows system DET plots when modeling is done using 1,2 and 3grams and a dictionary size of 64, with MAP adaptation (eq. 2, $\alpha = 0.01$). Results are encouraging as 1,2 and 3gram systems present an EER of 28, 30 and 31% respectively.

We next consider the effect on performance of this MAP adaptation on the acoustic event sequence system. Figure 3(b) shows a DET plot with results corresponding to a 2 and 3gram modeling method using a dictionary size of 256 proving the efficiency of the MAP method (EER goes from 36 to 26% and from 37 to 32% for 2 and 3-gram respectively).

6.2. Using the TFLLR method

Figures 3(c) 3(d) show the DET plots comparing the performance of the two scoring techniques (standard LLR and TFLLR eq. 5). A significant gain can be pointed out when using the TFLLR method (EER goes from 28 to 23% and from 30 to 27% for 2 and 3gram respectively).

6.3. Validation by breaking up features temporal order

The observed performance might be attributed to other factors than the acoustic event sequences. Two different experiments consisting in taking features in random temporal order were carried out to prove that the observed gain is due to the acoustic event sequences (experiments are performed using 2gram models, a dictionary size of 64, without MAP and TFLLR method).

- Applying a random process to training and test features: Figure 3(d) proves that the system performance is not due to the increase of the number of parameters to model speakers.
- Applying a random process to test features only: Figure 3(d) proves that information for speaker discrimination comes from acoustic event sequence modeling.
7. Fusion with a GMM ASR system

The purpose of this section is to prove that although information provided by acoustic event sequence systems is not sufficient for a stand-alone speaker detection system, it can be complementary to the information of a GMM system. Indeed, performance of these systems is low compared to a state-of-the-art GMM system (Figure 4). However, literature shows that performance is comparable with Ngram phone sequence based systems [4].

The GMM system presented in this section is the LIA system presented at NIST SRE 2004 without score normalization and the acoustic sequence systems are using TFLLR method with 64 symbols.

7.1. Correlation coefficients

Computation of the correlation coefficient is a good indicator of how complementary the information between two systems is. The correlation coefficient between a GMM system and our acoustic sequence modeling system is given in table 1. It seems that the correlation for target speakers is lower for longer Ngrams, taken less into account the 1gram influence which is closely related to the GMM.

<table>
<thead>
<tr>
<th>Ngram</th>
<th>Correlation Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-gram</td>
<td>0.619</td>
</tr>
<tr>
<td>3-gram</td>
<td>0.219</td>
</tr>
</tbody>
</table>

Table 1: Correlation coefficients between target and impostors scores of a GMM based system and a 2,3-gram based system.

7.2. Oracle fusion

Oracle fusion enables us to infer the optimal fusion between two systems. For a given decision point (in terms of FA/FR), the oracle gives the cases where fusion could have engendered performance. The EER point was chosen to achieve the analysis. There are 1320 target speaker tests, 14717 in total.

The purpose of this section is to prove that although information provided by acoustic event sequence systems systems is not sufficient for a stand-alone speaker detection system, it can be complementary to the information of a GMM system. Indeed, performance of these systems is low compared to a state-of-the-art GMM system (Figure 4). However, literature shows that performance is comparable with Ngram phone sequence based systems [4].

The GMM system presented in this section is the LIA system presented at NIST SRE 2004 without score normalization and the acoustic sequence systems are using TFLLR method with 64 symbols.

<table>
<thead>
<tr>
<th>Ngram</th>
<th>Discordance rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-gram</td>
<td>0.619</td>
</tr>
<tr>
<td>3-gram</td>
<td>0.219</td>
</tr>
</tbody>
</table>

Table 2: Oracle fusion. Discordance rate between the answers of the GMM (Acoustic) and the acoustic sequences system, percentage of right guess for both systems (number of tests).

7.3. Fusion Attempt

Regarding the previous experiments, a fusion attempt has been carried out by applying a non-weighted arithmetic mean to our scores. Scores were mapped beforehand onto the same score space by estimating means and covariance of both systems. The result is illustrated by a DET plot in figure 4 showing that a fusion of the GMM system and a 3gram based system brings a slight gain in the lower-right part of the DET curve as well as at the EER (0.8% of absolute win, 8% relative, to be compared to the 3% given by the oracle). This result is very promising, and confirms that the strategy adopted has to be studied deeper in order to get better results.

Figure 4: Unweighted arithmetic mean fusion DET curve

8. Conclusion

An innovative approach to model speakers by acoustic event sequences was shown. By building a speaker independent dictionary, and analyzing sequences of these symbols, speaker discrimination was performed. Experiments were carried out on relatively short duration signals on which high level feature approaches often fail. Moreover, applying a fusion with a state-of-the-art GMM verification system has shown to be promising. Integration of symbol duration and variable-length sequence modeling form the following work at this paper.

9. References