A Global Optimization Framework For Speaker Diarization

Mickael Rouvier, Sylvain Meignier

Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

A Global Optimization Framework For Speaker Diarization

Mickael Rouvier, Sylvain Meignier

LIUM, Université du Maine – Le Mans, France
{mickael.rouvier, sylvain.meignier}@lium.univ-lemans.fr

Abstract

In this paper, we propose a new clustering model for speaker diarization. A major problem with using greedy agglomerative hierarchical clustering for speaker diarization is that they do not guarantee an optimal solution. We propose a new clustering model, by redefining clustering as a problem of Integer Linear Programming (ILP). Thus an ILP solver can be used which searches the solution of speaker clustering over the whole problem. The experiments were conducted on the corpus of French broadcast news ESTER-2. With this new clustering, the DER decreases by 2.43 points.

1. Introduction

The goal of speaker diarization is to annotate temporal regions of audio recordings with speaker labels, in order to answer the question "who spoke when". This operation is performed without knowledge of the number of speakers or their identity. A common approach to this task consists in detecting homogeneous audio segments, which each contains the voice of only one speaker. The segments are then grouped into clusters, where each cluster contains segments of only one speaker.

Actually, in speaker diarization of broadcast news, the main methods of clustering are based on hierarchical, agglomerative algorithms such as top-down algorithms [1] or bottom-up algorithms [2]. Systems using the bottom-up approach (also known as Hierarchical Agglomerative Clustering – HAC) obtained the best results in the ESTER 2008 and NIST RT-04F evaluation campaigns. The HAC approach is an iterative algorithm that merges the two most similar clusters. This process is repeated until the similarity between any two clusters does not rise beyond some threshold value. Similarity is calculated using Gaussian Mixture Models (GMM). Unfortunately, greedy algorithms based on GMMs suffer from many problems.

First, the HAC approach is a greedy algorithm that solves the problem by choosing a local optimum at each step with the hope of finding a global optimum. However, during the greedy search, the selection of the next merge depends strongly on those chosen so far. An error at the beginning is propagated until the end of the clustering, causing an increase in the error rate.

Second, GMM-based speaker models convey not only useful information (related to the speaker) but also useless information that can disrupt the speaker clustering. This useless information can be of various nature and can be related to environment variability or channel-variability, for example.

In the work presented here, we propose a new clustering model where clustering is addressed as a global process – as opposed to the greedy approaches where it is treated as a series of local problems. We propose to replace the greedy, bottom-up search with a global formulation, where the basic bottom-up framework can be expressed as a variant of k-center problem. The algorithm can be expressed as a problem of Integer Linear Programming (ILP), through a definition of the concept of cluster in ILP terms. An ILP solver can then be used to minimize the result of the objective function.This new model is based on the i-vector paradigm. The i-vector approach was developed in an effort to enhance the classical speaker GMMs used in the field of Speaker Verification (SV), and was recently used in the field of speaker diarization of telephone conversations [3].

The paper is organized as follows: Section 2 presents the architecture of speaker diarization. Section 3 presents the corpus used for the experiments. Section 4 summarizes the i-vector approach. Section 5 presents our global optimization framework for speaker diarization. The results of our experiments are explained in Section 6. Section 7 then concludes with a discussion of possible directions for future works.

2. Architecture

The diarization system used is the LIUM Speaker Diarization system[4], freely distributed[1]. This system obtained the best results during the ESTER 2008 evaluation campaign.

The system is composed of an acoustic Bayesian Information Criterion (BIC)-based segmentation followed by a BIC-based hierarchical clustering. Each cluster represents a speaker and is modeled with a full covariance Gaussian. Viterbi decoding is used to adjust the segment boundaries using GMMs with 8 diagonal components for each cluster trained by Expectation-Maximization (EM).

[1] http://www-lium.univ-lemans.fr/diarization/

Segmentation, clustering and decoding are performed over 12 MFCC+E, computed with a 10ms frame rate. Music and jingle regions are removed using Viterbi decoding with 8 GMMs, for music, jingle, silence, and speech (with wide/narrow band variants for the latter two, and clean/noised/musical background variants for wide-band speech).

In the previous steps, features were used unnormalized in order to preserve information on the background environment, which may help differentiating between speakers. At this point however, each cluster contains the voice of only one speaker, but several clusters can be related to the same speaker. The contribution of the background environment to the cluster models must be removed through feature normalization in order to merge these clusters into one.

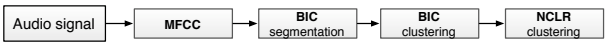

Figure 1: Architecture of the LIUM Speaker Diarization system

As shown in Figure 1, the system is completed with a Normalized Cross Likelihood Ratio (NCLR [5]) based on bottom-up clustering. It is performed on the clusters obtained after BIC segmentation: the parameters of each segment are normalized using feature warping and a Universal Background Model (UBM) is adapted (Maximum A Posteriori – MAP) for each cluster.

In this paper, we propose another method of clustering based on the i-vectors. We propose to substitute the last step of the system (the NCLR) with our new model. The previous steps (parameterization, segmentation...) remain the same.

3. Corpus

The data used for the experiments are those of the ESTER 2008 evaluation campaign [6]. The data were recorded from 4 French radio stations, and are divided into 3 corpora: the train corpus corresponds to more than 111 shows (90 hours), the development corpus corresponds to 20 shows, and the evaluation corpus contains 26 shows. The train corpus is employed to learn and to condition the i-vectors and the development corpus is employed to set the various thresholds of the systems.

4. i-vector

4.1. i-vector extraction

I-vector approaches have become the state-of-the-art in the SV field. They provide an elegant way of reducing a large-dimensional input data to a small-dimensional feature vector, while at the same time retaining most of the relevant information. The technique was originally inspired by the Joint Factor Analysis (JFA) framework introduced in [7].

Given a speaker- and channel-dependent GMM, the corresponding mean super-vector M can approximated by:

$$M = m + Tw \tag{1}$$

where m is the mean super-vector taken from a GMM-UBM; T is a low-rank rectangular matrix spanning the subspace covering the important variability; w is a low-dimensional vector with a normally distributed prior $N(0, \mathbf{I})$.

After iteratively estimating matrix T over a training corpus, equation 1 allows to use the lower-dimensional vector w as a speaker model in place of a large GMM. w is referred to as an i-vector.

The i-vector algorithm is fully described in [8].

4.2. i-vector conditioning and distance metric

4.2.1. Conditioning

At this step the i-vectors contain both speaker and channel information. The goal is to find a method that is able to carry out channel compensation. In [9, 10], the authors proposed to perform channel compensation in i-vector by using several channel compensation techniques working in this space. The best results were obtained by the process *LDA+WCCN+Fast scoring*.

But in [11], the authors propose a more robust method. That i-vector conditioning method is an iterative process with two goals.
i) Ensure that the i-vectors are distributed among $\mathbf{N}(0, \mathbf{I})$. One consequence of that constraint is that the vector dimensions of i-vectors are mutually independent.
ii) Apply length normalization to the i-vectors. In [12, 11], it is shown that length normalization made the test and trial i-vector distributions more similar and more Gaussian shaped.

In the training corpus, for each turn of speech obtained using the reference we extract an i-vector. The goal of the conditioning algorithm is to compute parameters for the i-vectors present in the training corpus and apply these parameters to the i-vectors present in the test corpus.

Algorithm 1 describes the training method for the i-vector conditioning parameters. The parameters (the mean μ_i and the covariance matrix Σ_i) of the i-vectors present in the train corpus are saved at each iteration i (*step 0*). Next, the i-vectors are conditioned using the parameters of the current iteration: *step 1* is the classical data standardization, and *step 2* is length normalization.

On the test corpus, after the BIC clustering, an i-vector is computed for each cluster. The i-vectors are then conditioned iteratively, in a manner similar to that used during the training phase, as explained in algorithm

Algorithm 1: Conditioning algorithm of i-vectors on the train corpus

for $i = 1$ *to* $nb\_of\_iterations$ **do**
  Step 0: Compute the mean μ_i and the covariance matrix Σ_i on the train corpus;
  for *each* w *in the train corpus:* **do**
    Step 1: $w = \Sigma_i^{-\frac{1}{2}} (w - \mu_i)$;
    Step 2: $w = \frac{w}{\|w\|}$;
  end
end

2. The difference lies in the absence of *step 0*: the mean μ_i and covariance matrix Σ_i used for each iteration in this phase are the ones saved during the training phase. As in the training phase, *step 1* is the data standardization, and *step 2* is length normalization.

Algorithm 2: Conditioning algorithm for the test phase

for $i = 1$ *to* $nb\_of\_iterations$ **do**
  Step 1: $w = \Sigma_i^{-\frac{1}{2}} (w - \mu_i)$;
  Step 2: $w = \frac{w}{\|w\|}$;
end

4.2.2. Distance metric

Given two i-vectors w_i and w_j computed for clusters i and j, the goal is to verify whether the two i-vectors correspond to the same speaker or not. If we assume homoscedasticity (equality of covariances) and Gaussian conditional density models, the most likely class can be obtained by the Bayes optimal solution:

$$d(w_i, w_j) = (w_i - w_j)\, W^{-1}\, (w_i - w_j)' \qquad (2)$$

where W is the within-class covariance matrix calculated on the conditioned i-vectors of the train corpus. So according to equation 2, the shorter the distance is, the more likely the two i-vectors belong to the same speaker. The within-class covariance is calculated :

$$W = \frac{1}{n} \sum_{s=1}^{S} \sum_{i=1}^{n_s} (w_i^s - \bar{w_s})\, (w_i^s - \bar{w_s})' \qquad (3)$$

where n_s is the number of utterances for speaker s, n is the total number of utterances, w_i^s is the training i-vector of session i for speaker s, and $\bar{w_s}$ the mean i-vector for speaker s over all the sessions.

5. Global Optimization Framework

The solution of speaker clustering given by HAC algorithms do not guarantee to provide an optimal solution. We propose to write our problem of speaker clustering in terms of ILP (Integer Linear Programming).

The goal of clustering is to group N i-vectors into K clusters (where K is to be determined by the algorithm and is of course between 1 and N). To transform our problem in an ILP, we use the hypothesis that i-vector n can belong to cluster k if the distance between the center of the cluster and the i-vector is less than a set threshold. For our problem, the center of a cluster is necessarily an i-vector. Theoretically, there may be as many clusters as i-vectors. The goal is to minimize the number of clusters, so that all i-vectors are assigned to a cluster.

From this, we can formulate the objective function and constraints of our problem. The objective function is to minimize the number of clusters, K, but also to minimize the dispersion of the i-vectors within every cluster. We define two binary variables y_k and $x_{k,n}$. The binary variable y_k indicates whether cluster k is selected. The binary variable $x_{k,n}$ indicates whether i-vector n belongs to cluster k. Thus our objective function can be written as:

$$z = \sum_{k=1}^{N} y_k + \frac{1}{F} \sum_{k=1}^{N} \sum_{n=1}^{N} d(w_k, w_n) x_{k,n} \qquad (4)$$

The objective function is decomposed in two parts: the first part ($\sum_{k=1}^{K} y_k$) calculates the number of clusters in our problem; the second ($\sum_{k=1}^{K} \sum_{n=1}^{N} d(w_k, w_n) x_{k,n}$) calculates the sum of the distances between the center of cluster k and the i-vectors attached to that cluster; where $d(w_k, w_n)$ is the distance between the center of cluster k and i-vector n. The resolution of this problem aims at minimizing both the number of clusters and dispersion. F is a normalization factor, to weight the two subparts of Equation 4.

We note that under our assumptions, the center of any cluster is in reality an i-vector, therefore calculating the distance between cluster k and i-vector n is a distance calculation between two i-vectors.

Thus, the speaker clustering model can be written as:

$$\begin{aligned}
&\text{Minimize} && z \\
&\text{Subject To} && \sum_{n=1}^{N} x_{k,n} = 1, && \forall k, (5) \\
& && x_{k,n} - y_k \leq 0, && \forall k, \forall n, (6) \\
& && d(w_k, w_n) x_{k,n} \leq \delta, && \forall k, \forall n, (7) \\
& && x_{k,n} \in \{0, 1\}, && \forall k, \forall n \\
& && y_k \in \{0, 1\}, && \forall k
\end{aligned}$$

Equation 5 ensures that all i-vectors have been assigned to a cluster. Equation 6 ensures that if an i-vector n is assigned to a cluster k, then the cluster k is selected. In

Equation 7, an i-vector n can be selected from a cluster k if the distance is lower or equal to distance δ. $d(w_k, w_n)$ corresponds to the distance given by Equation 2 between i-vector n and cluster k.

6. Results and comparison

6.1. i-vectors and ILP

Matrix T of equation 1 is estimated over the train corpus. The matrix is iteratively estimated using the Expectation Maximization (EM) algorithm. We used 60-dimensional acoustic features, with a 10ms frame rate, composed of 19 MFCCs plus log energy and augmented by first and second-order deltas. The GMM-UBM is a gender- and channel-independent GMM composed of 1024 Gaussians computed using the ALIZÉ speaker recognition toolkit[2].

In order to have a balance between the modeling precision and the amount of the data leading to accurate parameters estimation, we have chosen a dimension of 60 for the i-vectors. In fact, if we take an upper dimension for some segments, we cannot have a sufficient number of frames to correctly estimate matrix T.

The ILP clustering algorithm is developed using the GNU Linear Programming Kit[3].

6.2. Results

The default set of distance metric thresholds (δ) was determined using the ESTER 2008 development corpus. We can observe in Figure 2 that the optimal threshold on the development and test corpora is the same ($\delta = 180$). Preliminary results show that the threshold is practically the same for corpora of various kinds (meeting, TV). We may theorize that as we remove useless information (channel-effect, ...), the threshold focuses on speaker information and apparently there is less need to adapt it for different tasks.

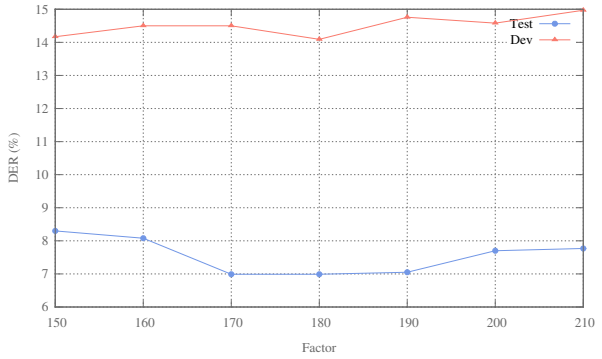

Figure 2: *Results on ESTER 2008 dev and test corpus, for different distance metric threshold.*

[2]http://alize.univ-avignon.fr/
[3]http://www.gnu.org/s/glpk/

We propose, following the same clustering algorithm (HAC), to compare NCLR (*NCLR HAC*) and i-vector (*i-vector HAC*). Then, we propose to apply the ILP model on i-vector (*i-vector ILP*). The system *NCLR HAC* is the classical system used for the campaign ESTER 2008. The results are reported in Table 1.

Table 1: Baseline system vs i-vectors (DER on evaluation corpus)
NCLR HAC: the baseline system
i-vector HAC: the system using the i-vectors and the HAC clustering
i-vector ILP: the system using the i-vectors and the ILP clustering

Corpus	NCLR HAC	i-vector HAC	i-vector ILP
Africa 1	9.60%	6.05%	2.79%
Inter	9.23%	11.72%	8.62%
RFI	3.61%	2.33%	2.33%
TVME	13.31%	13.17%	13.54%
ESTER-2	9.42%	9.08%	6.99%

The i-vectors with a HAC gives better results than the baseline but worse than the i-vectors with ILP clustering. The ILP algorithm explores more clustering solutions than the greedy HAC algorithm and the measure between i-vector is a distance whereas it is a similarity for NCLR. The i-vector ILP system obtains on TVME radio a worse result than the baseline (13.54% and 13.31% respectively). Most of the speakers (56%) use a phone in the shows from TVME radio. The GMM-UBM is gender- and channel-independent; we think that lay be the reason behind this result.

7. Conclusion

In this paper, we proposed a new model of speaker clustering based on i-vectors. The ILP, in place of standard NCLR-based clustering, obtains a DER decrease of 2.43 points on the test corpus of the ESTER 2008 evaluation campaign. The i-vectors give more robust models than GMMs for this task, as was already the case for speaker verification.

8. Acknowledgments

The authors would like to thank Pierre-Michel Bousquet for the help and the discussion about the conditioning algorithm of i-vector.

This research was supported by ANR (French National Research Agency) under contract number ANR-2010-CORD-101-01 (SODA project).

9. References

[1] Corinne Fredouille and Grégory Senay, "Technical improvements of the E-HMM based speaker di-

arization system for meeting records," in *MLMI*, Steve Renals, Samy Bengio, and Jonathan G. Fiscus, Eds. 2006, vol. 4299 of *Lecture Notes in Computer Science*, pp. 359–370, Springer.

[2] Claude Barras, Xuan Zhu, Sylvain Meignier, and Jean-Luc Gauvain, "Multi-stage speaker diarization of broadcast news," *IEEE Transactions on Audio, Speech & Language Processing*, 2006.

[3] Stephen Shum, Najim Dehak, Ekapol Chuangsuwanich, Douglas Reynolds, and Jim Glass, "Exploiting intra-conversation variability for speaker diarization," in *Interspeech*, 2011.

[4] Sylvain Meignier and Teva Merlin, "LIUM SpkDiarization: An open-source toolkit for diarization," in *CMU SPUD Workshop*, 2010.

[5] Viet-Bac Le, Odile Mella, and Dominique Fohr, "Speaker diarization using normalized cross likelihood ratio," in *Interspeech*. 2007, pp. 1869–1872, ISCA.

[6] Sylvain Galliano, Guillaume Gravier, and Laura Chaubard, "The ESTER 2 evaluation campaign for the rich transcription of French radio broadcasts," in *Interspeech*. 2009, pp. 2583–2586, ISCA.

[7] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[8] Najim Dehak, Patrick Kenny, Réda Dehak, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 99, pp. 1–23, 2010.

[9] Najim Dehak, Réda Dehak, Patrick Kenny, Niko Brümmer, Pierre Ouellet, and Pierre Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Interspeech*. 2009, pp. 1559–1562, ISCA.

[10] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[11] Pierre-Michel Bousquet, Driss Matrouf, and Jean-François Bonastre, "Intersession compensation and scoring methods in the i-vectors space for speaker recognition," in *Interspeech*, 2011.

[12] Daniel Garcia-Romero and Carol Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, 2011.