Optimizing feature representation for speaker diarization using PCA and LDA

Itshak Lapidot, Jean-Francois Bonastre

University of Avignon, LIA, 339 Chemin des Meinajaries BP 91228, Avignon, 84911 France
itsikv@netvision.net.il, jean-francois.bonastre@univ-avignon.fr

Abstract

In this work we examine the interest of both LDA and PCA applied on the mel-cepstrum coefficients for speaker diarization. PCA is applied before the diarization process when LDA is used after an initial diarization step. We show that PCA allows a reduction in diarization time but do not offer a diarization error reduction contrarily to LDA which allows a performance improvement of about 14.8% (relative).

1. Introduction

Speaker diarization is a process of segmenting a speech record into homogeneous segments, such that each segment will contain one speaker only. Additional type of segments, e.g., non-speech and overlap speech, should also be segmented. All segments from the same speaker have to be clustered together. Usually the number of speakers are unknown, however, in several application, like telephone conversations, this information is available [1]. In this work the diarization is performed on two speaker telephone conversations assuming that two speakers are present in the recordings.

This work aims to evaluate the potentiality of *principal-component-analysis* (PCA) and *linear-discrimination-analysis* (LDA), applied on the *mel-frequency cepstrum coefficients* (MFCC) for speaker diarization. Both PCA and LDA allow a data dimensionality reduction and a better conditioning of the data space, when LDA adds discriminant aspects to this data space. Thanks to these techniques, we hope to improve the speaker diarization performance, both in terms of error rates and in terms of computing resources.

We are using a speaker diarization baseline system based on an iterative approach. When iterative approach is applied, change detection and clustering are performed at once, and the process iterate until convergence. Our baseline system is an HMM-based system with fix duration constraint and with *self-organizing-map* (SOM) as emission probability estimator [2].

The PCA is applied on the MFCC as a data pre-processing step when LDA is applied after a preliminary diarization process. It is due to the discriminant nature of LDA, which needs to label the data for the training phases. We hope that after the preliminary diarization step, the speakers are sufficiently well separated to produce, thanks to LDA, a discriminative sub-space useful for additional speaker diarization iterations. PCA and LDA are also evaluated jointly.

The experimental part of this work is done on LDC America CallHome database [3] and uses NIST performance measure [4].

2. Fix duration diarization system

Our diarization system corresponds to the baseline system in [2]. Its block diagram is given in figure 1. At first 12 MFCC features from $20ms$ windows are extracted each $10ms$. A simple energy-based *voice activity detection* (VAD) is applied in order to obtain an initial speech/non-speech segmentation. In parallel, an overlap speech detection is performed. This process is based on maximum *a-posteriori* estimator of the wave form entropy, which is estimated each $100ms$. The overlap speech is pruned out of the conversation. The speaker models are initialized using *weighted-segmental K-means* (WSKM). The diarization is performed by 3-hyper-states HMM, corresponding to the two speakers and the non-speech. Each hyper-state model has 20 tied states ($200ms$) with transition probability 1 for the first 5 iterations of the diarization system and only 10 tied states ($100ms$) for the last iteration. In the rest of the paper, such training will be denoted as $L+1$ iterations, when L indicates the number of iterations with 20 tied states. The state models are SOM of size 6×10 used as log-likelihood estimators. Each code-word is assumed to be a mean of a Gaussian with identity covariance matrix. The transition matrix is initially trained using the initial segmentation received from the VAD and the WSKM. Following the training step, a Viterbi decoding is applied and gives a new segmentation and clustering.

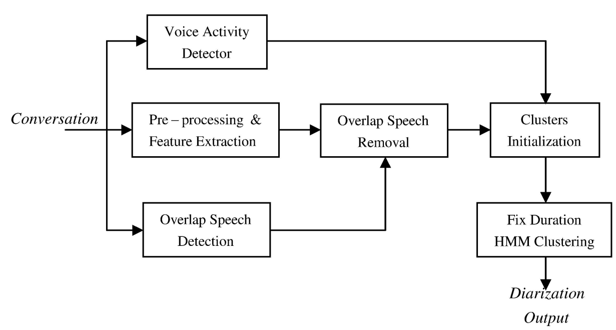

Figure 1: *Baseline diarization system.*

2.1. Overlap speech detection

The overlapping speech detector is described in [5]. The detection is performed in time domain. As the conversation is sampled with $16bits$, we calculate the probability of each quantization level. Be $x \in \mathcal{X}$ the quantization level random variable, and probability mass function (*pmf*) is $p(q_i) = Pr\{x = q_i\} = \frac{\#\{x_n : x_n = q_i\}}{N}$, when q_i is the quantization level; $\#$ is the number of samples of this level in the conversation; and N is the total number of samples in the conversation. After estimating *pmf*, the entropy of each segment, $100ms$ in length, is estimated given $p(q_i)$. For each segment $\{s_i\}_{i=1}^{S}$, the empirical entropy $\{E_i\}_{i=1}^{S}$ is given by $E_i = \frac{1}{N_i}\sum_{x_j \in s_i} \log\{x_j\}$, when N_i is the number of samples in each segment (in our case it is always $N_i = 800$, $100ms$ with $8kHz$ sampling rate).

After the computation of all the entropies, a one dimensional GMM with 4 mixture components is estimated over $\{E_i\}_{i=1}^{S}$. The Gaussian component with the smallest mean is referred to non-speech model when the component with the largest mean indicates overlapping speech and the other two, the two speakers. A MAP estimator is applied between the overlapping speech Gaussian component and the component of the speaker with the higher mean to find the minimum classification error threshold. However, it is not the best threshold to minimize the diarization error. This threshold was found to be relative to the MAP estimator threshold and the highest mean, as described in [5]. All the segments which are labeled as overlapping speech are discard from the diarization process.

2.2. Initialization using weighted segmental K-means

The WSKM algorithm was first presented at [6], and it is based on a variant of a standard K-means, the weighted K-means (WKM) algorithm. After the speech/non-speech separation, the mean of each speech segment, $\{S_i\}_{i=1}^{M}$, is calculated, $\{\mu_i\}_{i=1}^{M}$, when M is the number of speech segments at the output of the VAD. The weights associated of each segment are the length of the segment $\{w_i = N_i\}_{i=1}^{M}$. The WKM is applied on the $\{(\mu_i, w_i)\}_{i=1}^{M}$. The WSKM has the following algorithm:

1. Initiate the codebook with random selection of K vectors, out of $\{\mu_i\}_{i=1}^{M}$ (in our case $K = 2$), and the codewords are $\{W_k\}_{k=1}^{K}$.
2. For each code-word, find the segment means such that the cluster $C_k = \{\mu_i : \forall l \neq k \bullet d(\mu_i, W_k) \leq d(\mu_i, W_l)\}$, when $d(x, y)$ is an Euclidean distance.
3. Update the codewords as follows: $W_k^{new} = \frac{\sum_{\mu_i \in C_k} w_i \mu_i}{\sum_{\mu_i \in C_k} w_i}$ and assign $\{W_k^{new}\}_{k=1}^{K} \rightarrow \{W_k\}_{k=1}^{K}$.
4. If termination conditions met, exit; otherwise, return to step 2.

2.3. Emission likelihood estimation

For the state emission likelihoods, we use *self organizing maps* (SOM) [7]. Self-organizing map is a neural network that produces a stochastic vector quantization (VQ) which minimizes the mean squared error.We apply here the two phases training algorithm described in [1], [8]. The trained SOM neurons are the codewords of the codebook, $\{wc^l\}_{l=1}^{L}$. The assumption we made for the state emission log-likelihood estimation is that each code-word is the mean of a Gaussian with identity covariance matrix [9]. For each feature vector the state emission log-likelihood is the maximum of the log-likelihoods of one feature vector, x_n:

$$\begin{aligned} L(x_n|cw^{l^*}) &= -\tfrac{d}{2}\log(2\pi) - \tfrac{1}{2}(x_n - cw^{l^*})^t(x_n - cw^{l^*}) \\ cw^{l^*} &= \min_{l=1,\ldots,L}\{(x_n - cw^l)^t(x_n - cw^l)\} \end{aligned} \tag{1}$$

when t is the transpose operator.

2.4. Fix-duration HMM

An example of a 2-sates fix-duration HMM is shown in figure 2. Each time the system enters one hyper-state, it will stay at this state for a predefined number of frames, τ. When the system is in the last state (of an hyper-state) it will transit to the first state of any hyper-state (including returning to the first state of the current hyper-state). Inside the hyper-state, all the states are tied with the same emission probability model (same SOM) and with probability one to move to the next state. At each iteration, the best path is found using Viterbi search. A given hyper-state corresponds to one of the two speakers or to non-speech cluster. If it is not the final iteration, the models are retrained using standard SOM training algorithm, when the initial SOM for the retraining is the current SOM. The hyper-states transition matrix is also retrained, based on the Viterbi statistics.

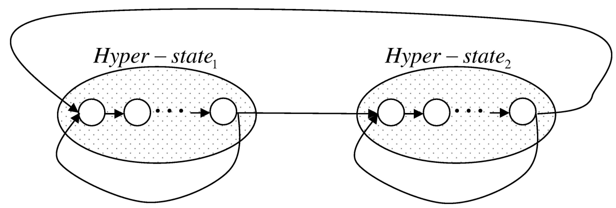

Figure 2: *2 States fix duration HMM.*

The transition matrix is composed of $K \times K$ blocks:

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1K} \\ A_{21} & A_{22} & \cdots & A_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ A_{K1} & A_{K2} & \cdots & A_{KK} \end{pmatrix} \tag{2}$$

Each block A_{qk} is a $\tau \times \tau$ matrix. (τ is the number of states in each hyper-state). Each matrix on the main diagonal is the transition matrix of moving inside hyper-state, and it contains only zeros and ones below the main diagonal, except the last element at the first raw which gives the probability of returning to the first state of the hyper-state (self-loop at the hyper-state level) eq. 3. The blocks out of the main diagonal contain only zeros except the last element at the first raw, which is the transition probability to move to hyper-state q from hyper-state k, eq. 4.

$$A_{kk} = \begin{pmatrix} 0 & \cdots & 0 & p(k|k) \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{pmatrix} \in \mathbb{R}^{\tau \times \tau} \tag{3}$$

$$A_{qk} = \begin{pmatrix} 0 & \cdots & 0 & p(q|k) \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{\tau \times \tau} \tag{4}$$

The probabilities at the place $(1, \tau)$ are trained by the Viterbi statistics.

3. PCA and LDA encapsulation

In this work we evaluate the influence of the PCA applied before the diarization and the influence of LDA applied after several diarization system iterations. Figure 3 presents the block diagrams of the two processes.

As we said before, labeled data is required for LDA. So, we apply 5 iterations of the diarization system, as we observed that the diarization system usually converges to a local maximum after 5 iterations of the algorithm. Then, we apply LDA using the obtained labels, which correspond to non-speech, speaker 1 and speaker 2. Finally, five more iterations of Viterbi clustering and system retraining are performed using the projected data

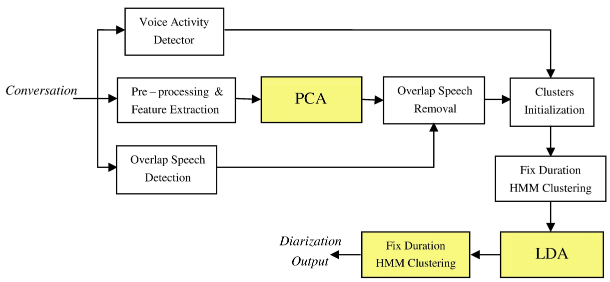

Figure 3: *PCA-LDA-based diarization system.*

with minimum duration of $200ms$, followed by a final iteration with minimum duration of $100ms$.

The LDA-based diarization could be applied on the MFCC features but also on MFCC features following by PCA. It gives a total of three variants to compare verse the baseline system:

1. PCA: $5 + 1$ iterations of the diarization system, applied on MFCC features preprocessed by PCA
2. LDA: 5 iterations of the diarization system, followed by LDA, and additional $5 + 1$ iterations on the LDA-tranformed features.
3. PCALDA: 5 iterations of the diarization system, applied on MFCC features preprocessed by PCA. The LDA is performed and the system is launched for $5+1$ additional iterations using the resulting features (where PCA and LDA were sequentially applied).

The experiment results obtained by these systems are presented in the following section.

4. Experiments and results

This section describes the database used for the experiments; the performance measure; the *diarization error rate* (DER) and the experiments which are conducted.

4.1. Database

For evaluation purposes, 108 conversations extracted from LDC America CallHome English language corpus were used [3]. It corresponds to all the conversations with an associated transcription. In most of the conversations, about 10 minutes are transcribed and were used. The data was sampled at $8kHz$ in a 2 channel μ-law format. The two channels were summed to generate a two speaker conversation.

4.2. Diarization error rate (DER)

Diarization Error Rate was defined by NIST in order to evaluate the speaker diarization task [4]. The DER is defined by:

$$DER = 100 \cdot \frac{\sum_{s=1}^{S} \{dur(s)(\max(N_{Ref}(s), N_{Sys}(s)) - N_{Correct})\}}{\sum_{s=1}^{S} \{dur(s) \cdot N_{Ref}(s)\}} \quad (5)$$

where $dur(s)$, $N_{Ref}(s)$, $N_{Sys}(s)$ and $N_{Correct}(s)$ correspond to the duration of the segment s; the number of speakers assigned to segment s; the number of speakers assigned by the system to segment s; and the number of speakers assigned by the system to segment s which actually takes part in s, respectively.

In the DER computation, $0.5sec$ of speech around the changing points is excluded to the scoring applied, i.e., $0.25sec$ of speech on each side of each changing point is not used for the scoring.

4.3. PCA-based experiment

Figure 4 shows the results of the PCA experiment. As we used 12 MFCC features, we can project the data between 1 to 12 projection directions; the last case corresponds to a rotation of the features in the 12 dimensional space. We can see that there is no improvement in terms of DER but the results are quite similar for 9 PCs and more, compared to the baseline.

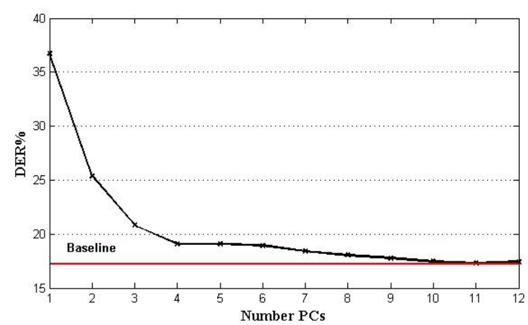

Figure 4: *DER as a function of the number of PCs.*

Figure 5 shows the runtime of the diarization system as a number of selected principal components (PCs). It can be seen that the time changes almost linearly with the number of PCs. This result was expected as the most time consuming part of our system is the SOMs' training phase, and it has a linear dependence on the feature vector dimension. The overhead for PCA calculation in negligible. The overall runtime time of the diarization system for 108 conversations with a PCA order of 12 (rotation only) is about $2Sec$ longer compared to the baseline system time (about $1739Sec$).

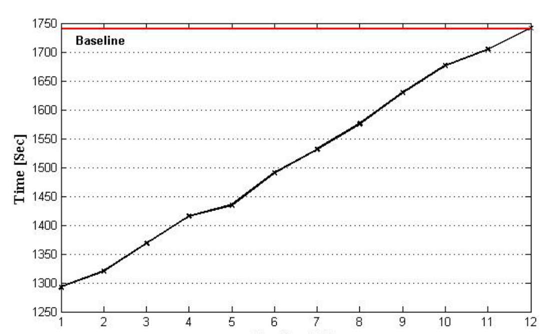

Figure 5: *Diarization time as a number of PCs.*

4.4. LDA-based experiment

Figure 6 (black line) shows the results of the LDA experiment, where the LDA is applied alone. As in the first experiment, the

MFCC features are of dimension 12, so we could use between 1 to 12 projection axes; again, the last case is just rotation of the features in the 12 dimensional space. Of course, the runtime needed by this system is higher than the baseline's one, as additional 5 iterations are required. To assess the performance of our LDA system, we are using here a $10 + 1$ version of the baseline system. This version obtains a DER of 16.72%, to be compared with 14.25% for our best LDA system (using projection on 5 linear discriminant vectors). It corresponds to a relative improvement of about 14.77% of the DER.

4.5. PCA-LDA-based experiments

For the third experiment, we combine PCA and LDA. For the PCA, we select between 9 and 11 PCs (these dimensions correspond to the ones which obtained similar DER compared to the baseline system). Figure 6 shows the results of this experiment. It presents the obtained DER depending on the number of selected PCs and the number of LDA dimensions. For all PCA setups, the best number of LDA components is 4 or 5, which is consistent with the second experiment. The obtained DER are always better than the baseline $10 + 1$ system DER but remain slightly worse than the performance obtained by the LDA taken alone. However, a small reduction of runtime can be achieved in the first stage as it is shown in figure 5.

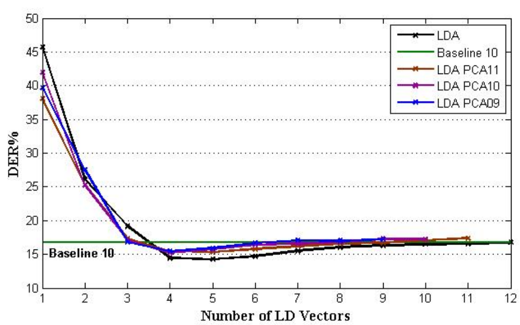

Figure 6: *DER as a function of the number of PCs and LD vectors. The black line is corresponds to the LDA only system.*

5. Conclusions

In this work we explored the potential of PCA and LDA within the framework of 2-speaker telephone conversations speaker diarization, using an iterative diarization system. The LDA approach allows a relative DER reduction of about 15%. It demonstrates the interest of this discriminant feature transformation for speaker diarization, knowing that the needed information about the classes to discriminate is obtained fully automatically by running the diarization system itself before to apply the LDA. By comparison, the PCA seems not able to improve the discriminant power of the feature space as no DER improvment was observed using PCA, versus the baseline system. Nethertheless, PCA allows a (relatively small) reduction of diarization runtime (6.3% for 9 PCs) without any loss in terms of DER, still compared to the baseline system. Combining both PCA and LDA also allows better performance compared to the baseline system but do not improve the results of the LDA taken alone (except a small reduction of the system runtime).

If the interest of a discriminant transformation of the feature for speaker diarization was demonstrated in this paper, several questions remain open. First, the LDA relies on the quality of the labeling used to train it. This quality depends both on the speech recording itself and on method used to obtain the first labeling.

Concerning the speech recordings, we applied our LDA approach on NIST SRE-05 [10] and we were unable to observe a performance improvement using LDA. The average segment length of LDC America CallHome database is about $2.07sec$ when it is less than 1 one second for NIST SRE-05. For LDC America CallHome database, the initial diarization process produces good enough results to estimate the LDA discriminant transformation and, finally, good performance is obtained using LDA. For NIST it is not the case, due according to us to the average segment duration, and LDA was not able to produce a good discriminant space. This remark shows the dependency between the intrinsic difficulty of the speech recordings and the usefulness of our LDA approach. Nevertheless, we used our baseline diarization system optimized in terms of diarization performance in order to obtain the initial labeling used to train the LDA parameters. It seems interesting to us to optimize this step specifically for our objective: produce robust class informations in order to train LDA parameters. Our hypothesis is that a task oriented process will allow to both improve the LDA performance improvement and the LDA robustness depending on the speech recordings. Finally, other discriminant approaches could also be investigated.

6. References

[1] O. Ben-Harush, O. Ben-Harush, I. Lapidot, and H. Guterman, "Initialization of iterative-based speaker diarization system for telephone conversations," *IEEE Trans. on Audio, Speech, and Languge Processing*, vol. 20, no. 2, pp. 414 – 425, Feb. 2012.

[2] I. Lapidot and J.-F. Bonastre, "Generalized viterbi-based models for time-series segmentation applied to speaker diarization," in *ODYSSEY 2012 -The Speaker and Language Recognition Workshop*, 2012 accepted.

[3] "Liguistic data consortium," LDC97S42, Catalog, 1997, available: http://www.ldc.upenn.edu/Catalog.

[4] "Nist diarization criterion," available: http://www.itl.nist.gov/iad/mig/tools/.

[5] O. Ben-Harush, I. Lapidot, and H. Guterman, "Entropy based overlapped speech detection as a pre-processing stage for speaker diarization," in *Proceedings of Interspeech 2009*, 2009.

[6] ——, "Weighted Segmental K-Means Initialization for SOM-Based Speaker Clustering," in *Proceedings of Interspeech 2008*, 2008.

[7] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[8] I. Lapidot, H. Guterman, and A. Cohen, "Unsupervised speaker recognition based on competition between self-organizing maps," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 877–887, Jul. 2002.

[9] I. Lapidot, "SOM as Likelihood Estimator for Speaker Clustering," in *Proc. Eurospeech03*. Geneva, Switzerland: ISCA, September 1-4 2003, pp. 3001–3004.

[10] "National institute of standards and technology," The NIST 2005 Speaker Recognition Evaluation, 2005, available: http://www.itl.nist.gov/iad/894.01/tests/spk/2005.