EURECOM-BISC system description

Jose Patino, Héctor Delgado, Nicholas Evans

Department of digital security, EURECOM, Sophia Antipolis, France

{patino,delgado,evans}@eurecom.fr

Abstract

This document describes the systems submitted by EURECOM-BISC to the first DIHARD diarization challenge, which is based on a binary key (BK) modelling technique. BK-based approaches to speaker diarization are flexible in their use of background training data and can operate entirely without training data, learning necessary background information at runtime. By making no assumptions about the domain, this particular quality of BK-based approaches to diarization, make it particularly well suited to domain-robust diarization.

1. Data resources

The systems submitted do not use any external training data. No additional database was used. Only the DIHARD development set was used to tune system parameters. The systems work at a per-file basis, i.e. the required models are estimated directly on the audio stream being processed.

2. Detailed description

5 systems were submitted. They essentially share the same structure and core technology, but vary on particular modules like the acoustic frontend employed, speaker estimation and clustering. All the variants of each module are described next. Then each system is defined as a combination of modules.

2.1. Acoustic frontend

Two different acoustic frontends were used. As part of our baseline, we employed traditional Mel-frequency cepstral coefficient (**MFCC**) features [1], comprising 19 static coefficients computed from windows of 25ms with 10ms shift and with a filterbank of 20 channels. Additionally, and as one of the proposed enhancements, we employed a frontend called *infinite impulse response - constant Q, Mel-frequency cepstral coefficients* (**ICMC**) [2]. These features have been successfully applied to tasks like speaker recognition, utterance verification [2] and speaker diarization [3]. These features are similar to MFCC, but they replace the short time Fourier transform by an infinite-impulse response, constant Q transform (IIR-CQT) [4]. This is a richer, multi-resolution time-frequency representation for audio signals, which provides a greater frequency resolution at low frequency and a higher time resolution at high frequency. These ICMC features use longer windows of 128ms with 10ms shift.

2.2. Segment representation

Speech segments and speaker clusters are represented by means of the binary key speaker modelling. This technique was initially proposed for speaker recognition [5, 6] and applied to speaker diarization [7], voice activity detection [8] and emotion recognition [9]. The binary key representation has been well investigated for speaker diarization [8, 10, 7]. The principal goal of that series of works was to perform fast, efficient speaker diarization with no need of external training data, while maintaining a competitive performance. The method represents speech segments as low-dimensional, speaker-discriminative binary or integer vectors, which can be then clustered using similarity measures. The core model to perform this mapping is a binary key background model (KBM) which is trained in the test segment before diarization. The KBM is actually a collection of diagonal-covariance Gaussian models selected from a pool of Gaussians learned on a sliding window of duration 2s over the test data. The window rate is adjusted dynamically to assure a minimum of 1024 Gaussians. Then, a selection process is performed to keep a percentage $p = 10\%$ of the Gaussians in the pool to assure a good coverage of all the speakers in the test audio stream by following the selection procedure described in [7].

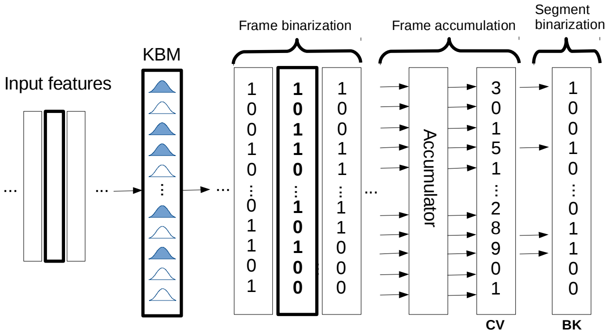

Figure 1: *Diagram of the cumulative vector (CV) and binary key (BK) estimation procedure.*

The KBM can be then used to map a sequence of acoustic features into a sparse, speaker-discriminative D-dimensional binary or integer vector, D being the size of the KBM, as described in Figure 1. The procedure starts with a feature-level binarisation. The i-th feature vector is evaluated against each Gaussian in the KBM, producing a vector of likelihoods. Then, the likelihood vector is mapped to a binary vector by setting to 1 the N_G positions of the top-N_G likelihood in the likelihood vector. This procedure is repeated for every feature vector in the segment, obtaining a binary matrix of top-Gaussians per feature. The next step is the accumulation of the binary matrix by column-wise summation, resulting in a cumulative vector (CV) which accounts for the number of times that each Gaussian in the KBM has been activated (i.e. has produced an N_G-top likelihood) for the input feature sequence. The last step is a segment-level binarisation by selecting the top M positions to one while keeping the rest to zero. The obtained BK contains the information of *which* Gaussians better represent the input feature sequence. Full details of the algorithm can be found in [7]. Submitted systems use $N_G = 5$ and keeps the CV as the final representation. CVs are estimated on the test stream using a sliding window of 3s with a shift of 1s.

2.3. Speaker estimation

The estimation of the number of clusters is based on two different algorithms. For the baseline, the selection is performed using an **elbow criterion** which is applied to the curve of the within-class sum-of-squares (WCSS) of all clustering solutions, with the goal of finding a trade-off between the number of clusters and cluster dispersion, as presented in [10].

A proposed alternative is based on the **spectral clustering** algorithm described in [11][1]. This algorithm first estimates the number of clusters and then performs the clustering. In this section the speaker estimation part is explained. Given a test audio file, represented by a sequence of segment CVs, the affinity matrix (matrix of pair-wise segment similarities) is calculated using the cosine similarity and refined by a series of operations, including Gaussian blur with standard deviation σ, row-wise thresholding of similarities below the p-percentile, symmetrisation, diffusion and row-wise Max normalisation (consult [11] for full details). The refinement process smooths and denoises the data in the similarity space. Then, eigenvalue decomposition is performend on the processed affinity matrix. Being $\lambda_1 > \lambda_2 > ... > \lambda_n$ the decreasingly sorted eigenvalues, the number of clusters $\widetilde{k}$ is selected as the value k which maximises the eigengap, defined as follows:

$$\widetilde{k} = \arg \max_{1 \leq k \leq n} \frac{\lambda_k}{\lambda_{k+1}} \tag{1}$$

Since the most frequently larger eigengap tends to be the first one λ_1/λ_2, the algorithm returns 1 cluster for many files. For this reason we forced the algorithm to return 2 or more clusters by excluding the first eigengap. On the other end, the maximum number of speaker is restricted to 10 (this was decided based on the maximum number of speakers seen on the development set). Finally, having found that diarization errors in single-speaker documents produce high error rates, we designed a specific mechanism for single-speaker detection. Single-speaker detection is then performed according to the thresholding of the eigengap between the two largest eigenvalues. In the case that $\lambda_1 - \lambda_2$ exceeds a threshold θ, empirically set to 410, then the number of clusters is set to 1.

2.4. Clustering

Two different approaches are explored for the clustering modules of the submitted systems. On the one hand, and as described in [7], the **AHC clustering**, uses a bottom-up agglomerative clustering algorithm as follows. First, a number of clusters $N_{init} = 25$ is initialised by a uniform splitting of test data and their CVs estimated. Then the segment CVs are compared to the clusters' CVs through the cosine similarity and assigned to the closest one. If the number of clusters estimated by the clustering selection module has been reached, the current clustering solution is returned and the process stopped. Otherwise, the two closest clusters are merged and the clustering process repeated.

Alternatively, we experimented with the complete **spectral clustering** algorithm [11] explained in the previous section for the speaker estimation problem. Once the number of clusters $\widetilde{k}$ has been estimated, the $\widetilde{k}$ eigenvectors corresponding to the $\widetilde{k}$ largest eigenvalues are used as a new $\widetilde{k}$-dimensional representation of the input segments, which are then clustered with a k-means algorithm using the squared Euclidean distance.

2.5. Resegmentation

A resegmentation process is performed to refine time boundaries of the segments generated in the clustering step. It uses Gaussian mixture models (GMM) to model the clusters, and on maximum likelihood scoring at feature level. Since the log-likelihoods at frame level are noisy, an average-smoothing within a 1s sliding window is applied to the log-likelihood curves obtained with each cluster GMM. Then, each frame is assigned to the cluster which provides the highest smoothed log-likelihood.

2.6. System definition

The systems submitted were numbered in a sequential manner as additional modules were added, ranging from 1, for our baseline, to 5, for our best performing system. The different systems are then defined by their particular implementations of the acoustic frontend, clustering, and speaker estimation modules. All of them use the same resegmentation module at the end of the pipeline. The systems are defined as follows:

- **SYSTEM_1**: Based on the system described in [7], uses a MFCC frontend, followed by AHC clustering, with the number of speakers being estimated through the use of the elbow criterion. The relative KBM size is $p = 85\%$. This setting was used in prior work and taken as a baseline configuration.
- **SYSTEM_2**: Equivalent to SYSTEM_1, but employing the proposed ICMC [2] frontend and a relative KBM size of $p = 10\%$. The new setting was found after a grid search on the KBM size using ICMC features on an oracle experiment where the optimum number of clusters was chosen arbitrarily (i.e. the clustering solutions which minimised the diarization error rate as a per-file basis).
- **SYSTEM_3**: This system derives from SYSTEM_2 and replaces the AHC clustering and the WCSS-elbow based speaker estimation with the spectral clustering solution described in Sections 2.3-2.4
- **SYSTEM_4**: It combines AHC clustering of SYSTEM_2 with the spectral speaker estimation module of SYSTEM_3.
- **SYSTEM_5**: equivalent to **SYSTEM_4**, this system includes the single-speaker detection module described in Section 2.3.

2.7. Hardware requirements

Submitted system do not require any special hardware. A regular desktop PC can be used. The execution times shown in Table 1 were calculated on a desktop computer equipped with an Intel i5-4440 CPU @ 3.10GHz, and 16 GB RAM. The systems were implemented and run in Matlab R2017b using a parallel pool of 4 workers to take advantage of all CPU cores. Table 1 shows execution times taken to each system to process the file "DH_0024.wav" of the development set, measured in seconds as as the real time factor (xRT), that is, the execution time divided by the duration of the audio file. We show xRT of the complete

[1] Accepted for publication in ICASSP 2018. A draft is available in `https://arxiv.org/abs/1710.10468`

Table 1: *Execution time to process the file "DH_0024" of the development, measured in seconds and as the total real-time factor (xRT) and when excluding feature extraction.*

Module	time (s)	xRT	xRT (no feat. extract.)
SYSTEM_1	11.24	0.029	0.019
SYSTEM_2	5.94	0.059	0.010
SYSTEM_3	2.95	0.054	0.005
SYSTEM_4	5.91	0.059	0.010
SYSTEM_5	6.01	0.059	0.010

system and excluding feature extraction. Since ICMC feature extraction is more computationally demanding than MFCC extraction, xRT for SYSTEM_2-SYSTEM_5 are higher than for SYSTEM_1. However, if we exclude the feature extraction stage, it can be seen that SYSTEM_2-SYSTEM_5 are more efficient than SYSTEM_1. This is due to the use of a smaller KBM size of $p = 10\%$ compared to SYSTEM_1 which uses a baseline KBM size of $p = 85\%$. This parameter dictates the dimension of the segment and cluster CV representations, which has an impact on system speed. It is also seen that SYSTEM_3 using the spectral clustering algorithm is the most efficient one.

3. References

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[2] H. Delgado, M. Todisco, M. Sahidullah, A. K. Sarkar, N. Evans, T. Kinnunen, and Z. H. Tan, "Further optimisations of constant q cepstral processing for integrated utterance and text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2016, pp. 179–185.

[3] J. Patino, H. Delgado, N. Evans, and X. Anguera, "Eurecom submission to the albayzin 2016 speaker diarization evaluation," in *Proc. IberSPEECH*, Nov 2016.

[4] P. Cancela, M. Rocamora, and E. López, "An efficient multi-resolution spectral transform for music analysis," in *Proceedings of ISMIR*, 2009, pp. 309–314.

[5] X. Anguera and J.-F. Bonastre, "A novel speaker binary key derived from anchor models." in *Proc. INTERSPEECH*, 2010, pp. 2118–2121.

[6] G. Hernandez-Sierra, J. R. Calvo, J.-F. Bonastre, and P.-M. Bousquet, "Session compensation using binary speech representation for speaker recognition," *Pattern Recognition Letters*, vol. 49, pp. 17 – 23, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865514001779

[7] H. Delgado, X. Anguera, C. Fredouille, and J. Serrano, "Fast single-and cross-show speaker diarization using binary key speaker modeling," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 12, pp. 2286–2297, 2015.

[8] H. Delgado, C. Fredouille, and J. Serrano, "Towards a complete binary key system for the speaker diarization task." in *Proc. INTERSPEECH*, 2014, pp. 572–576.

[9] J. Luque and X. Anguera, "On the modeling of natural vocal emotion expressions through binary key," in *Proc. 22nd European Signal Processing Conference (EUSIPCO)*, 2014, pp. 1562–1566.

[10] H. Delgado, X. Anguera, C. Fredouille, and J. Serrano, "Novel clustering selection criterion for fast binary key speaker diarization," in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 3091–3095.

[11] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *Proc. ICASSP*, Calgary, Canada, 2018.