Towards a complete Binary Key System for the Speaker Diarization Task

Héctor Delgado[1], *Corinne Fredouille*[2], *Javier Serrano*[1]

[1]CAIAC, Autonomous University of Barcelona, Spain
[2]University of Avignon, CERI/LIA, France

hector.delgado@uab.cat, corinne.fredouille@univ-avignon.fr, javier.serrano@uab.cat

Abstract

Speaker diarization is the task of partitioning an audio stream into homogeneous segments according to speaker identity. Today state-of-the-art speaker diarization systems have achieved very competitive performance. However, any small improvement in Diarization Error Rate (DER) is usually subject to very large processing times (real time factor above one), which makes systems not suitable for some time-critical, real-life applications. Recently, a novel fast speaker diarization technique based on speaker modeling using binary keys was presented. The proposed technique speeds up the process up to ten times faster than real-time with little increase of DER. Although the approach shows great potential, the presented results are still preliminary. The goal of this paper is to further investigate this technique, in order to move towards a complete binary-key based system for the speaker diarization task. Preliminary experiments in Speech Activity Detection (SAD) based on binary keys show the feasibility of the binary key modeling approach for this task. Furthermore, the system has been tested on two different kinds of test data: meeting audio recordings and TV shows. The experiments carried out on NIST RT05 and REPERE databases show promising results and indicate that there is still room for further improvement.

Index Terms: speaker diarization, binary key, speech activity detection

1. Introduction

Speaker diarization is the task of segmenting an audio file into speaker-homogeneous segments. Currently speaker diarization has become a very common pre-processing tool for many speech-related tasks which take advantage of dealing with speech signals from a single-speaker. That is the case, for instance, of speech recognition, which can be improved through speaker adaptation. Media accessibility projects dealing with speech technologies to provide access to audiovisual content could also benefit from speaker diarization as a pre-processing tool for the subsequent speech technologies. Furthermore, searching speech utterances spoken by target speakers within big audiovisual content repositories is increasingly becoming very popular and challenging. Before identifying such speakers by means of speaker identification technology, they must be previously separated adequately. Here, speaker diarization systems should be accurate and fast enough in order to process big quantities of data in a reasonable time period.

Most state-of-the-art systems rely on the use of Gaussian Mixture Model (GMM) as speaker models, trained using maximum likelihood or discriminative training approaches. Bayesian Information Criterion (BIC) is usually used to decide which cluster pairs should be merged, as well as a stopping criterion. Finally, data assignment is done by means of Viterbi decoding. All the mentioned algorithms are applied iteratively, imposing a high computational load which results in too long processing times [1] (above 1xRT, being xRT the Real Time factor) for some real-life applications.

Some efforts have been done in order to face the problem of speed. [2] optimizes some parts of an agglomerative clustering algorithm, resulting in an execution time of 0.97xRT. [3] Reported real time factors up to 0.008xRT by parallelizing the GMM training algorithms using GPU. The first approach seems not to be fast enough, while the second option is very dependent on complex, non-standard hardware architectures.

Recently, a novel speaker diarization framework was proposed in [1], based on the “binary key” speaker modeling described in [4]. This diarization system runs over 10 times faster than real time with performance just slightly above a baseline acoustic-based system. DER scores of around 27% with a real time factor of 0.103 xRT were reported using all the NIST RT databases. This technique provides a fast alternative to the GPU approach but using a single CPU. In consequence, this solution can be easily portable across more standard platforms.

This work follows the direction towards a complete binary key system for the speaker diarization task. For this purpose, the binary key framework is tested to perform SAD in order to check its feasibility for this task. In addition, a switch from meeting room recordings to TV broadcast data is done, as recently this kind of data is increasingly gaining more attention. Therefore, SAD and speaker diarization experiments are performed using both meeting audio and TV audio using several configurations to evaluate performance. These preliminary experiments show that the binary key SAD approach outperforms a “classic” HMM-based audio segmentation tool using the NIST RT05 database. In addition, when switching to TV broadcast data, speaker diarization performance remains quite similar with minimum system adaptation to the new domain.

The paper is structured as follows: Section 2 describes the binary key speaker diarization system. Section 3 proposes binary key approach for SAD. Section 4 describes the experimental setup and results. Section 5 concludes and proposes future work.

2. Speaker diarization using binary keys

The implementation of the binary key diarization system is based on the system described in [1]. An overall description is given in the current section (refer to [1] for further details). As shown in figure 1, two different parts can be distinguished. First, the acoustic processing block aims at transforming the acoustic input data into a suitable binary representation. Secondly, the binary processing block takes the binary data from the previous stage to perform an agglomerative clustering but, unlike the classic approach, all the operations are performed in

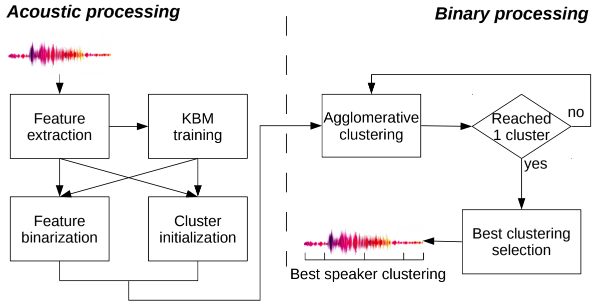

Figure 1: *Overview of the binary key based speaker diarization system.*

the binary domain, which results in a significant gain in execution time, compared with state-of-the-art agglomerative systems.

2.1. Acoustic processing block

As said above, this block transforms the acoustic feature vectors into binary vectors called binary keys. The key element for this transformation is a UBM-like acoustic model, called KBM (binary Key Background Model), which is trained using the own test input data, but in a particular way. A single Gaussian is trained every n seconds (with some overlap), so that in the end a pool of several hundreds of Gaussians is obtained. Proceeding in this way, it is guaranteed that the overall acoustic space is covered by the pool of Gaussians. The next step consists in taking a subset of N components from the pool so that the selected Gaussians are as complementary and discriminant between them as possible. To achieve that, the Gaussians are selected iteratively by calculating the KL2 (symmetric Kullback-Leibler) divergence between the already selected components and the remaining ones, and the most dissimilar component is selected. The process is repeated until having N components.

Once the KBM is trained, any set or sequence of input feature vectors can be converted into a binary key. A binary key $\mathrm{v}_f = \{v_f[1], ..., v_f[N]\}, v_f[i] = \{0, 1\}$ is a binary vector whose dimension N is the number of components in the KBM. Setting a position $v_f[i]$ to 1 (TRUE) indicates that the ith Gaussian of the KBM coexists in the same area of the acoustic space as the acoustic data being modeled. The binary key can be obtained in two steps. Firstly, for each feature vector, the best N_G matching Gaussians in the KBM are selected (i.e., the N_G Gaussians which provide higher likelihood for the given feature), and their identifiers are stored. Secondly, for each component, the count of how many times it has been selected as a top component along all the features is calculated. Then, the final binary key is obtained by setting to 1 the positions corresponding to the top M Gaussians at the whole feature set level, (i.e., the Mth most selected components for the given feature set). Intuitively, the binary key keeps the components of the KBM which best fit data being modeled, preserving only the ones with highest impact. Note that this method can be applied to any set of features, either a sequence of features from a short speech segment, or a feature set corresponding to a whole speaker cluster. This fact will make the comparison between two binary keys straightforward, either between segment-cluster key pairs or cluster-cluster key pairs.

The last step before switching to the binary process block is the clustering initialization. This is done at the acoustic level in order to have an initial rough clustering as a starting point. Taking advantage of the KBM trained before, an initial set of N_{init} clusters is build by using the first N_{init}th Gaussians in the KBM. The input data are divided into small segments (e.g., 100ms) and they are assigned to the cluster which Gaussian provides the highest likelihood.

2.2. Binary processing block

The binary block implements an agglomerative clustering approach. However, all operations are done with binary data, what makes the process faster than with classic GMM-based approaches. First, binary keys for the initial clusters are calculated using the method explained in section 2.1. Then, the input data are reassigned to the current clusters. Data are first divided into fixed length segments and binary keys are calculated for all them. Note that these binary keys will be used along the iterations of the agglomerative clustering, so they can be stored and reused. Next, the segments are assigned by comparing their binary keys with all current cluster binary keys. The similarity metric is given by equation 1.

$$S(\mathrm{v}_{f1}, \mathrm{v}_{f2})) = \frac{\sum_{i=1}^{N}(v_{f1}[i] \wedge v_{f2}[i])}{\sum_{i=1}^{N}(v_{f1}[i] \vee v_{f2}[i])} \qquad (1)$$

where $\wedge$ indicates the boolean AND operator, and $\vee$ indicates the boolean OR operator. This is a very fast, bit-wise operation between two binary vectors.

Once data are redistributed, binary keys are trained for the new clusters. Finally, similarities between all cluster pairs are obtained using equation 1 and the cluster pair with the highest score is merged, reducing the number of clusters by one.

The iterative process is repeated until a single cluster is reached, storing all the partial clusterings. At the end of the process, the final clustering is output by using a modification of the T-test T_S metric proposed in [5]. After the computation of intra-cluster and inter-cluster similarity distributions between segments for each clustering C^i, the selected clustering is the one which maximizes T_S, given by equation 2.

$$T_s = \frac{m_1 - m_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \qquad (2)$$

where m_1 , σ_1 , n_1 , m_2 , σ_2 and n_2 are the mean, standard deviation and size of intra-cluster and inter-cluster distance distributions, respectively.

3. Binary key based SAD

The binary key speaker modeling has shown great potential to discriminate between speakers [4]. Even the technique has been successfully applied in emotion recognition as well [6].

Given the underlying GMM-UBM-like nature of the framework, it could seem reasonable to hypothesize that the binary key modeling could also be suitable for other audio classification tasks (such as SAD), which are usually addressed using GMMs.

In this section, a binary key based approach to SAD is proposed. As for speaker modeling, a KBM model is needed in order to capture the overall acoustic space. Then, binary keys for the desired audio classes have to be obtained. Finally, data assignment will be done by comparing binary keys of segments of input data with acoustic classes binary keys.

In the case of SAD, the KBM is trained in a different way. First, a GMM is trained for each acoustic class (e.g., “speech” and “nonspeech”) using Expectation-Maximization (EM) algorithm with appropriate labeled external data. The final KBM is the result of concatenating all Gaussian components of the individual GMM. Therefore, a KBM build from two 16-component GMM will contain 32 Gaussian components.

Once the KBM is obtained, binary keys for each class can be trained with labeled data by using the binary key computation method explained in section 2.1.

Finally, the input data are assigned to the various acoustic classes. The input signal is split into equal-sized small segments. Binary keys are computed for all them, and they are compared with the binary keys of all audio classes. Each segment is assigned to the class for which binary key maximizes the similarity measure given by equation 1.

4. Experiments and results

This section describes experimental setups and results for two different tasks. First, the proposed SAD algorithm is evaluated. Secondly, the obtained SAD labels are used as input labels for the binary key speaker diarization system in order to discard non speech content. Additionally, the system is also tested using SAD labels obtained from a standard HMM-based SAD system for comparison. Finally, some execution time figures are shown.

The two tasks are evaluated under two conditions: meeting room data, and broadcast TV data. For meeting room experiments, the NIST RT05 dataset is used, whilst the REPERE [7] phase 1 test dataset is used for TV audio experiments. The NIST RT05 database consists of a set of 10 meeting excerpts. In the case of this paper, the Multiple Distant Condition (MDM) of the NIST RT evaluations is used. Regarding the TV data, the REPERE database was developed in the context of the REPERE Challenge [8]. It consists of a set of TV shows from several French TV channels. For speaker diarization performance comparison, refer to the NIST RT05 [9] and REPERE [10] evaluations results.

4.1. Experimental setup

Experiments on both kinds of audio data (meeting room and TV broadcast) share a common experimental setup, except some aspects regarding audio channel handling and feature extraction, which are specified next. In the case of meeting audio experiments, the multiple audio channels for each meeting are first filtered through a Wiener filter to reduce noise, and then a single, enhanced channel is obtained using beamforming [11]. For TV audio, the provided single channel is used without further treatment. Next, feature extraction is performed. In the case of speaker diarization, standard 19-order MFCCs are computed using a 25ms window, every 10ms. However, for SAD experiments, 12-order LFCCs augmented with energy and first and second derivatives (totaling a vector of 39 elements) are used.

For training the KBM for speaker diarization, single Gaussian components are estimated using a 2s window in order to have sufficient data for parameter estimate. Window rate is set according to the input audio length, in order to obtain an initial pool of 2000 Gaussians. In the case of the KBM for the SAD task, the KBM is estimated following the method explained in section 3 by using GMMs for each audio class.

With regard to binary key estimate parameters, the top 5 Gaussian components are taken in a frame basis, and the top 20% of the components at segment level.

The clustering initialization is done by using the first 16 Gaussian components in the KBM as cluster models for meeting audio experiments. In the case of TV data, the number of initial clusters is augmented to 25, as the database contains audio files with higher number of speakers (up to 18 in some excerpts). Then, 100ms segments are assigned to the different clusters to obtain the first rough, over-segmented clustering.

Table 1: *SAD results using the NIST RT05 and REPERE as test data. The segmentation error is broken down into miss speech and false alarm. Baseline results using HMM-based SAD are also included for comparison*

	NIST RT05			REPERE		
KBM comp.	Miss	False alarm	Seg. error	Miss	False alarm	Seg error
64	5.4	1.9	6.46	-	-	-
128	4.4	1.7	6.13	8.52	**0.93**	9.47
256	3.1	1.9	**4.97**	6.13	**1.01**	7.14
512	3.1	1.7	**4.85**	5.66	**1.1**	6.76
Classic SAD	4.5	1	5.47	1.73	2.35	4.08

Finally, in the agglomerative clustering stage, binary keys are computed for each 1s segment, augmenting it 1s before and after, totaling 3s.

In order to evaluate performance, the output labels are compared with the reference ones to compute the DER. Since the proposed system does not handle overlap speech, regions with more than one active speakers are ignored in the score computation (note that this is only for evaluation, so that overlapped speech regions are included during the complete diarization process).

4.2. Binary key based SAD results

As mentioned in section 3, the KBM estimate for SAD is performed in a different way as it is done for speaker diarization. This is due to the need of adequate labeled training data to estimate the audio classes models. Given that labeled data for the test signal are normally unavailable, here the KBM is trained using external data.

For the experiments on meeting audio, two audio classes are used (“speech” and “nonspeech”), whilst 4 different classes (“speech”, “speech plus music”, “music”, and “telephone”) are utilized for TV audio.

The data assignment is done in a similar way as in the diarization system. But given the nature of the audio been classified, the window length should be significantly smaller than the speaker window. For instance, using windows of 1s would not capture pauses shorter than this. It has been established in the literature that the minimum pause length to be taken into account should be 0.3s. It is for this reason that here the analysis window is set to 0.3s.

Table 1 shows performance of the proposed binary key based SAD approach using meeting room audio and TV audio. In addition, results of an HMM-based [12] approach are given for comparison.

In experiments on the NIST RT05 database, the proposed SAD outperforms the baseline SAD when using 256 and 512 components in the KBM. However, the average false alarm error keeps higher and this can have an additional impact as noise is being introduced as speech content. With regard to the REPERE database, the binary key SAD performance is slightly worse when using a 512 component KBM. However, the false alarm error remains lower for all the performed tests.

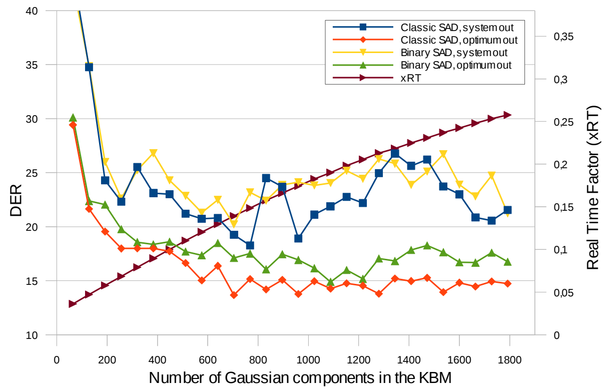

Figure 2: *Speaker diarization results on the NIST RT05 dataset using HMM-based and binary-key-based SAD labels*

4.3. Binary key based speaker diarization results

Figure 2 shows the DER trend according to N, N being the number of components in the KBM (note that N is also the number of bits in the binary key) using the NIST RT05 database as test data. The experiment is performed with both SAD labels from the classic HMM-based and the binary-key based approaches. Furthermore, DER scores for two different outputs are shown: the DER of the clusterings returned by the system, and the DER of the optimum clusterings selected manually (e.g., clusterings for which DER is minimum). This is done to evaluate the effectiveness of the final clustering selection algorithm. To measure execution times, the real time factor xRT is calculated, as the ratio between the time the system takes to perform the diarization (excluding SAD and feature extraction) and the time labeled as speech by the SAD system.

DER results clearly show that the system stopping criterion is not returning the optimum clusterings. In fact, DER of system output oscillate between 20% and 27% when the number of components is incremented, whilst the DER of the selected optimum clusterings converges around 15% when using HMM-based SAD labels, and around 17% when using binary-key SAD labels.

Although the binary key SAD system outperformed the HMM-based SAD system in previous SAD experiments, the resulted DER when they are used within the diarization process is affected negatively. An increment of around 2-3% absolute can be appreciated with respect to the use of the HMM-based SAD labels. There are two factors that can affect the final DER of the system. First, the average false alarm error rate of the binary SAD system is slightly higher than for the HMM-based SAD. Since false alarm errors introduce noise to the speech signal being processed, clusters become less pure. This may result in weaker speaker models. And second, it has been observed that the label boundaries are not as precise as the HMM-based system. This is due to the use of fixed-length segments when assigning data to an acoustic class, which does not allow to finely adjust the beginning and end of each segment.

Regarding the number of components in the KBM, DER converges for N above 500 (for optimum clusterings manually selected). From this point, incrementing N does not result in a DER improvement. Respecting execution time, xRT increases linearly when incrementing N. For $N = 512$, xRT is around 0.11. The best result of 13.66% DER, is achieved using $N = 704$, with 0.14 xRT.

Figure 3 shows the DER trend according to the number of components in the KBM for the case of TV audio. Note that, except the initial number of clusters (25 initial clusters versus 16 for meeting audio experiments), the experimental setting is exactly the same for both meeting and TV data experiments. Taking this into account, system performance is not far from the case of meeting audio experiments. Once again, the weakness of the optimum clustering selection criterion is reflected. However, when manually selecting the optimum clusterings, the system shows performance only slightly worse than in the case of meeting audio. With regard to the used SAD labels, DER varies around 2% absolute when using SAD labels obtained with both HMM and binary keys. The reason of this decrease could be the higher error rate of the binary key SAD system (around 2.7% absolute higher than the baseline HMM-based system). In addition, the problem of segment refinement mentioned above can have an added, negative impact.

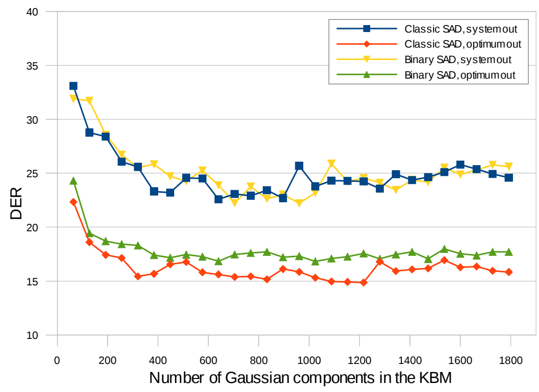

Figure 3: *Speaker diarization results on the REPERE dataset using HMM-based and binary-key-based SAD labels*

5. Conclusions and future work

This work focuses on the binary key based speaker diarization approach and aims at augmenting it by following two lines: the binary key based SAD task, and the use of TV broadcast data. SAD experiments show the potential of the binary key modeling for such audio classification problem. In addition, when switching to TV broadcast data, speaker diarization performance remains quite similar with minimum system adaptation to the new domain. However, the final clustering selection algorithm does not return the optimum clustering, so it has to be revised in order to apply the system to real cases. Apart from that point, all the above shows the feasibility of the binary key based speaker diarization and SAD for processing big repositories of TV shows. It is thought that an in-deep analysis of the KBM training parameters will allow to tune the system according to the nature of data being processed (audio duration, number of speakers, speaker turn length, etc), leading to a significant gain in performance. Finally, refining the obtained SAD labels boundaries will correct the small differences in DER when using the classic SAD labels.

6. Acknowledgments

This work is part of the project “Linguistic and sensorial accessibility: technologies for voiceover and audio description”, funded by the Spanish Ministerio de Economía y Competitividad (FFI2012-31023). This work was partially done within the French Research program ANR Project PERCOL (ANR 2010-CORD-102). This article is supported by the Catalan Government Grant Agency Ref. 2009SGR700. The authors would like to thank Xavier Anguera for his valuable support during the development of the system.

7. References

[1] X. Anguera and J.-F. Bonastre, "Fast speaker diarization based on binary keys," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 4428–4431.

[2] Y. Huang, O. Vinyals, G. Friedland, C. Muller, N. Mirghafori, and C. Wooters, "A fast-match approach for robust, faster than real-time speaker diarization," in *Automatic Speech Recognition Understanding, 2007. ASRU. IEEE Workshop on*, Dec 2007, pp. 693–698.

[3] E. Gonina, G. Friedland, H. Cook, and K. Keutzer, "Fast speaker diarization using a high-level scripting language," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, Dec 2011, pp. 553–558.

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

[5] T. H. Nguyen, E. Chng, and H. Li, "T-test distance and clustering criterion for speaker diarization," in *INTERSPEECH*, 2008.

[6] X. Anguera, E. Movellan, and M. Ferrarons, "Emotions recognition using binary fingerprints," in *IberSPEECH*, 2012.

[7] A. Giraudel, M. Carré, V. Mapelli, J. Kahn, O. Galibert, and L. Quintard, "The repere corpus : a multimodal corpus for person recognition," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012.

[8] J. Kahn, O. Galibert, L. Quintard, M. Carre, A. Giraudel, and P. Joly, "A presentation of the repere challenge," in *Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on*, June 2012, pp. 1–6.

[9] J. Fiscus, N. Radde, J. Garofolo, A. Le, J. Ajot, and C. Laprun, "The rich transcription 2005 spring meeting recognition evaluation," in *Machine Learning for Multimodal Interaction*, ser. Lecture Notes in Computer Science, S. Renals and S. Bengio, Eds. Springer Berlin Heidelberg, 2006, vol. 3869, pp. 369–389.

[10] O. Galibert and J. KahnAude, "The first official repere evaluation," in *Proceedings of the First Workshop on Speech, Language and Audio in Multimedia (SLAM 2013)*, Marseille, France, 2013.

[11] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2021, September 2007.

[12] C. Fredouille, S. Bozonnet, and N. W. D. Evans, "The LIA-EURECOM RT'09 Speaker Diarization System," in *RT 2009, NIST Rich Transcription Workshop, May 28-29, 2009, Melbourne, USA*, Melbourne, UNITED STATES, 05 2009.