Universitat Autònoma de Barcelona

Doctoral Thesis

Fast Cross-Session Speaker Diarization

Author:
Héctor Delgado Flores

Advisors:
Dr. Javier Serrano García
Dr. Xavier Anguera Miró

PhD Program in Electrical and Telecommunication Engineering

Department of Telecommunication and Systems Engineering

June 2015

Abstract

Today, massive amounts of audiovisual content are being generated, stored, released and delivered, in part due to the virtually unlimited storage capacity, the access to the necessary media to produce them by anybody and anywhere, and the ubiquitous connectivity provided by the Internet. In this context, suitable, affordable and sustainable content management which enables searching and retrieving information of interest is a must. Since manual handling of such amount of data is intractable, it is here where speech processing techniques may play a crucial role in the automatic tagging and annotation of audiovisual content.

The task of speaker diarization (also known as the "who spoke when" task) has become a key process as a supporting technology for further speech processing systems, such as automatic speech recognition and automatic speaker recognition, used for the automatic extraction of metadata from spoken documents.

Among the massive amount of audiovisual content being created, there can be recurrent speakers who participate in several sessions within a collection of audiovisual sessions. For instance, in TV and radio content one can frequently find recurrent speakers such as public figures, journalists, presenters, anchors, and so on. Due to the local nature of current speaker diarization technology (systems work on a single-session basis), an arbitrary recurrent speaker will likely receive different local abstract identifiers among the different sessions where he/she participates. In this situation, it would be more meaningful that the recurrent speakers receive the same global, abstract ID along all sessions. This task is known as cross-session speaker diarization.

Current state-of-the-art speaker diarization systems have achieved very good performance, but usually at the cost of long processing times. This limitation on execution time makes current systems not suitable for large-scale, real-life applications, and becomes even more evident in the task of cross-session speaker diarization.

In this thesis, the fast speaker diarization approach based on binary key speaker modeling is taken to a next level with the aim of bringing it closer to state-of-the-art performance while preserving high speed rates that enable the processing of large audio collections in competitive times. Furthermore, a new cross-session speaker diarization system based on binary key speaker modeling is proposed by following the same previously established goals: competitive performance with short execution times. As a result of this thesis, we propose a new improved single-session speaker diarization system which exhibits a 16% relative improvement in performance with regard to a baseline binary key system (15.15% DER opposed to 18.22% DER, being DER the diarization error rate), while being 7 times faster (0.035xRT against 0.252xRT, being xRT the real-time factor) and 28 times faster than real time. As for cross-session speaker diarization, in this thesis we propose a binary system whose performance is just slightly below (3.5% absolute DER) the performance of its single-session counterpart, while presenting a real-time factor of 0.036xRT. Furthermore, our approach has been shown to successfully scale for processing audio collection of several hundreds of hours.

Acknowledgments

First, I want to thank my advisor, Dr. Javier Serrano for his guidance and support along this thesis work, and also for giving me the opportunity to participate in very interesting projects during these last years at UAB.

There are two people that I would especially like to express my gratitude. The first one is my second advisor Dr. Xavier Anguera. I still marvel, not only at his invaluable and insightful technical support, but also at his exceptional generosity and encouragement shown towards me. The second one is Dr. Corinne Fredouille, who extraordinarily guided and supported me during my fruitful stay at the University of Avignon, and also through countless e-mailing from then on. Probably, this PhD thesis would have not been possible without their help. Many thanks to you both.

I also want to thank Dr. Jordi Carrabina for giving me the chance to join CEPHIS lab, where this adventure began.

I am very grateful to Dr. Pilar Orero and Dr. Anna Matamala from the Transmedia Catalonia research group for their willingness to help me as many times as I needed.

Thanks to the guys who have shared this time with me during all these years at UAB: Rosa Herrero, Adriana Ramírez, José David Rojas, Alejandra Ruvalcaba, Catya Zuñiga, Eduard Céspedes, Aitor Rodríguez, Marc Moreno, Roger Puig, Carlos Montero, Juan Carlos Chak, Óscar Lopes, Guillermo Talavera, Màrius Montón, Borja Martínez, Xavier González, Víctor Montilla, Joan García, Marta Viana, Marc Codina, Jordi González, Theodora Tsapikouni, Ana Alcalde, Xavier Palmer, Benyamin Ahmadnia, Diego González, Estel·la Oncins, Katerina Tsaousi and Carla Ortiz.

Very special thanks to my partner, Helena, for her unconditional support, for her patience, for being by my side in the most stressful and hardest moments, and for enjoying the good ones. This work is yours.

Of course, I also want to express my gratitude to my beloved family members: my parents María Teresa and Fernando, and my siblings María Teresa, Fernando and Carmelo. To them, I dedicate this work. Thank you all.

I also have to thank Esperanza, Tomás, Alberto and Rocío for their constant support.

Finally, thanks to my mates of my rock band “Thesauros” Dani, Iván and José, because not everything in life is work, and they have helped me to evade and relax making music from time to time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introductory chapter presents the context and the motivations that brought us to develop this thesis work, and the proposed objectives to be achieved. Section 1.1 sets the context and the motivations of this work. Section 1.2 defines the thesis objectives. Finally, Section 1.3 gives an outline of the rest of this thesis manuscript.

1.1 Context and motivation

Speech is the natural way that humans use to communicate among themselves. Speech is not only used to directly understand and to interact each other in our everyday life, but also as a way to spread and share information registered and stored in form of speech. This is the case, for example, of audio or audiovisual content produced and delivered on radio and television broadcast, audiovisual content made available on the Internet, and many others.

The fact is that for years now, and due to a virtually unlimited storage capacity, massive spoken contents are being produced, stored and released. These contents are not only generated by the mass media but also by people: today, anybody has affordable access to the necessary equipment to generate multimedia content at home. This, added to the access to the Internet by practically everyone, contributes to the publication of hundreds of hours of audiovisual content per minute worldwide.

These massive amounts of data undoubtedly become a source of information of great value. However, no matter how much stored information one has if there are no mechanisms to discern which information is of interest. The naive solution would be the manual inspection of the content until finding the information of interest. It seem pretty obvious that this task is intractable by humans. In place, the discipline of document indexing and retrieval aims at facing this aspect when the information is stored in form of written text. This discipline is mature nowadays, and it is widely present on current Internet search engines, daily used by many people.

However, when the information is stored in form of recorded audio, written-document indexing and retrieval techniques cannot be directly applied. It is here, where speech processing may play an important role. More concretely, the technique of audio indexing may be used to automatically extract information from the audio signal. The extracted information may be stored in form of text, and therefore document indexing and retrieval could be performed on the extracted metadata. In this situation, we talk about spoken document indexing and retrieval.

Different speech processing techniques could be used in order to automatically extract information from audio content. Probably, the most powerful is Automatic Speech Recognition (ASR) for transcribing the speech registered in the recording. Speech is probably the most essential source of information within an audio stream since it contains the message, which will probably code the majority of information. But the transcription is not the only useful resource. In fact, other information units could increase the value of the speech transcription. For example, knowing "who said what" can be used as a useful complement to "what was said" since a sequence of words may have different interpretations depending on who pronounces them.

The task of audio diarization consists in annotating an input audio signal with information that attributes (possibly overlapping) temporal regions of signal to their specific sources [Reynolds and Torres-Carrasquillo, 2004]. These sources can include particular speakers, music, background noise sources, and other signal source/channel characteristics. According to the target application, those sources or classes can be defined with different granularity, ranging from general wider acoustic classes (e.g. speech and non-speech, background noise, music) to more specific (speakers involved). When the defined classes correspond to the identities of the different speakers involved in the recording, the task is referred to as speaker diarization. A common definition found in the literature is that of answering the question "who spoke when". More precisely, speaker diarization can be defined as the process of partitioning an input audio stream into speaker-homogeneous segments and grouping them according to the speaker identities. The output of a speaker diarization system comprises a set of temporal segments (ideally, each segment will contain speech from a single speaker), each one with an associated abstract ID which identifies the speaker who speaks in such segment. Note that the associated IDs do not correspond to actual identities, leaving this task to subsequent speaker identification systems if required.

The term speaker diarization was coined in the early 2000's in the context of the Rich Transcription (RT) evaluation campaigns promoted by the National Institute for Standards and Technology (NIST). However, research on this field started several years before referred to as the tasks of speaker segmentation and speaker clustering. In 1996, the NIST HUB-4 evaluation on speaker-independent speech recognition motivated further research in speaker segmentation and clustering in order to enable speaker adaptation of ASR acoustic models. In this context, the symmetric Kullback-Leibler metric was first proposed in [Siegler et al., 1997] for speaker segmentation and clustering tasks. Later, [Chen and Gopalakrishnan, 1998] introduced the Bayesian Information Criterion (BIC) for speaker segmentation and clustering for the first time. These are two of the most extended approaches to speaker segmentation and clustering still used nowadays, especially BIC.

Over the years, most of the research in speaker diarization has been conducted on three main domains. First, speaker diarization was investigated on telephone recordings in the context of the NIST Speaker Recognition evaluations[1] in the late 1990's. Then the focus was put on the broadcast news domain, consisting in recordings of radio and TV broadcast programs. This domain was mainly promoted by the DARPA's (Defense Advanced Research Projects Agency) EARS program (Effective, Affordable, Reusable Speech-to-text) for rich transcription of broadcast news in 2002-2004. Finally, the focus switched to the meeting recordings domain, principally promoted by the CHIL[2] (Computers in the human loop) and AMI[3] (Augmented Multiparty Interaction) European projects in 2004.

Speaker diarization can be seen as a combination of the already mentioned tasks of speaker segmentation and speaker clustering. Speaker segmentation aims to find potential speaker change points within the audio stream, while speaker clustering aims to group those generated segments into speaker clusters. Generally, the task of speaker diarization assumes two premises:

- The number of speakers is unknown a priori.
- The identities of the speakers in the recording are unknown.

It is therefore an unsupervised task where it is the system's must to estimate the actual number of speakers and adequate speaker models. However, these premises are not mandatory in order to talk about speaker diarization. A priori knowledge about the actual number of speakers could be exploited in some situations. For example, in telephone calls recording, the number of speakers is very often 2, and this prior knowledge could be used, for example, in order to improve accuracy of a diarization system dealing with that kind of audio recordings.

Majority of current approaches to speaker diarization usually follow one of the two main strategies, namely the agglomerative clustering approach, and the divisive approach. The first (by far, the most adopted one) follows a bottom-up scheme where the number of initial clusters is greater than the number of speakers, and those clusters are iteratively merged until meeting a stopping criterion. Contrarily, the divisive approach follows a top-down strategy, starting with one single cluster which is iteratively split into smaller clusters, until meeting a stopping criterion. [Tranter and Reynolds, 2006] and [Anguera et al., 2012a] provide two excellent reviews of the most relevant speaker diarization approaches and algorithms developed from the beginnings to present.

With regard to its applications, speaker diarization is rarely used as a main technology, but as a supporting tool for further speech-related processing. For example, within a spoken document retrieval framework, speaker diarization can be used for speaker indexing. The extracted information of speaker turns can be incorporated to the speech-to-text transcriptions as additional metadata. This added value may enable the user to perform more precise queries. For example, the user could be interested in locating

[1] NIST Speaker Recognition evaluation website: `http://www.itl.nist.gov/iad/mig/tests/spk/`

[2] CHIL project website: `http://www.research.ibm.com/mcs/CHIL/chil_index.html`

[3] AMI project website: `http://www.amiproject.org/`

certain keywords occurrences, but only if they are pronounced by a target speaker within a spoken document or a set of spoken documents. But such speaker related information is not only useful for fetching information of interest, but also for facilitating the user experience when browsing within a piece of audiovisual content. In this situation, an enriched video player could provide direct access to the speaker turns, for example, by means of a temporal browsing bar.

Another very extended application of speaker diarization is as a supporting technology for ASR. In this scenario, the detected speaker turns may be used for adapting speaker-independent ASR acoustic models to target speakers who appear in the audio stream. In some way, speaker diarization enables the creation of profiles for target speakers. Combined with other audio indexing technologies, broader profiles could be estimated, for example, for female and male speech, and narrowband and broadband audio quality. The impact of speaker diarization within an ASR framework has already been studied by [Gauvain et al., 1999] in the transcription of broadcast news, and by [Stolcke et al., 2010] in the transcription of meeting recordings.

A third application of speaker diarization is to split multi-speaker signals into single-speaker signals which will be later processed by systems which usually deal with single-speaker audio excerpts. Some of these systems could be speaker tracking, speaker identification and verification systems. For example, in [Reynolds and Torres-Carrasquillo, 2004] speaker diarization is used to improve speaker recognition accuracy on conversational telephone speech.

Current speaker diarization systems are intended for working with isolated audio sessions. In this modality, each detected speaker within the session is assigned an abstract and local identifier. We refer to this modality as single-session speaker diarization. If a given speaker is present in more than one session, she/he will probably receive different abstract IDs among them. Recently, this fact motivated to extend the classic task of speaker diarization to the called cross-session speaker diarization task. In this new framework, speakers participating in different sessions must be identified with a collection-wise unique ID. The task of finding recurrent speakers within a collection of sessions has also been referred to as speaker linking [van Leeuwen, 2010] and speaker attribution [Ghaemmaghami et al., 2011]. This speaker diarization modality could be very useful in some domains of applications. For instance, in TV and radio content one can frequently find recurrent speakers such as public figures, journalists, presenters, anchors, and so on. Under these circumstances, it is highly desirable to identify each unique person with the same ID in a collection-wise fashion. This new modality has been promoted by the ETAPE[4] (*Évaluations en Traitement Automatique de la Parole*, addressed to speech technologies for spontaneous speech in TV streams) and REPERE[5] (*Reconnaissance de PERsonnes dans des Emissions audiovisuelles*, addressed to multi-modal person recognition in TV programs) evaluations, in 2011 and 2012, respectively.

Up to now, much work has been conducted in the field of speaker diarization, obtaining very competitive performance. However, there is still a number of open challenges which make this technology not totally usable in real life scenarios. One of them concerns

[4] ETAPE evaluation website: `http://www.afcp-parole.org/etape-en.html`

[5] REPERE evaluation website: `http://www.elra.info/en/projects/archived-projects/etape/`

speed when processing large amounts of data. With the increasing volume of audiovisual content, systems should be fast enough in order to process hundreds of hours in a reasonable time period. Current systems usually perform a combination of several accurate but costly algorithms applied in an iterative scheme. Commonly, a combination of Gaussian Mixture Models for speaker modeling, BIC for speaker segmentation and cluster merging, Viterbi decoding for data assignment, and others, are used to effectively perform speaker diarization, but at a cost of long processing times (xRT above 1, being xRT the Real-Time factor), what makes it very difficult to be suitable for real life applications where time is a key requirement. Some work has been done in order to speed up speaker diarization, but they rely on code parallelization and parallel hardware such as GPUs instead of trying to intrinsically reduce the complexity of the algorithms involved. This limitation also affects to the task of cross-session speaker diarization as it is highly dependent on the efficiency of single-session speaker diarization.

Recently, a novel speaker diarization framework was proposed in [Anguera and Bonastre, 2011], based on the "binary key" speaker modeling described in [Anguera and Bonastre, 2010]. The binary key speaker modeling relies on a special kind of Universal Background Model (UBM) which is used to convert a speech utterance into a single binary vector (only containing zeros and ones), called binary key, which retains speaker specific information. Both the training of the UBM and the estimation of the binary key are very fast processes. This fact motivated the inclusion of this speaker modeling into a speaker diarization framework. [Anguera and Bonastre, 2011] reported DER scores of around 27% with a real time factor of 0.103 xRT (around 10 times faster than real time) using all the NIST RT databases of meeting recordings. This technique provides a fast alternative, at the cost of a slight performance degradation. However, obtained results were considered still preliminary and we think that the technique has potential and that there is room for further improvement.

With regard to the task of cross-session speaker diarization, we also think that the binary key speaker modeling can be a suitable option for working with large amounts of data. Its capability of compacting long speech segments, or even complete speaker clusters, into a single vector of binary values makes it possible to transform large amounts of acoustic feature vectors into a reduced number of binary keys. This reduction of the input data may be a key point towards a tractable solution for the cross-session speaker diarization task on large data collections.

1.2 Thesis objectives

After analyzing the context and identifying the main issues, we can define the thesis objectives. The main goal of this thesis is to get a fast, yet accurate, cross-session speaker diarization system, suitable for processing big data collections with recurrent speakers. From this global objective, several specific objectives can be derived.

- Implementing a single-session speaker diarization system based on the binary key speaker modeling to be used as a baseline system. This system is to be based on the system described in [Anguera and Bonastre, 2011]. Once implemented, the system will be evaluated to check performance consistency with the original work.

- The baseline system will be analyzed in depth in order to identify those potential points whose modification could lead to improvements in both accuracy and execution time. Additionally, the binary key speaker modeling is also to be studied with the aim of getting insight on how the overall performance could be improved.

- Given the conclusions extracted from the previous study, modifications to the identified key points of the baseline speaker diarization system will be proposed, implemented, and assessed experimentally.

- Once the three objectives above are achieved, we will have obtained a fast and accurate single-session speaker diarization system. Then, a novel cross-session speaker diarization based on binary keys is to be proposed. This system will be based on the single-session system, and its mean features will also be high efficiency and accuracy. Then, the system will be evaluated experimentally.

We are interested in getting a very fast diarization system with competitive performance, without the requirement of external data. Our approach will neither require training nor development data at all; all the resources involved are trained on the test data itself. We are aware of the great performance achieved by modern speaker diarization systems based on advanced speaker modeling, such as Gaussian supervectors, speaker factor and i-vectors, but those approaches require vast amounts of training and development data in order to estimate Universal Background Models (UBM), total variability matrices, and so on. Those techniques are, therefore, out of the scope of this thesis.

As for the data domain, we opt for the broadcast news recordings. This is a domain in which the task of cross-session speaker diarization may be especially useful due to the frequent participation of recurrent speakers among a collection of sessions. In this thesis, all the experimental work is to be performed on the REPERE Phase 1 database which consists of excerpts extracted from different TV shows from the BFM TV, and LCP French channels. This choice will allow us to compare performances achieved in this thesis work with the official REPERE evaluation results, as well as with some other published works based on the same dataset.

1.3 Outline

In addition to this introductory chapter, this thesis is structured into five more chapters.

Chapter 2 provides a review of the state of the art in speaker diarization. After a brief introduction, the basic concepts and modules involved in classic speaker diarization are reviewed. Then, more recent research lines are examined, putting a special focus on the two main topics of this thesis: speeding up speaker diarization, and cross-session speaker diarization. Finally, the most used metrics for assessing speaker diarization performance are described.

Chapter 3 first describes the original approach of binary key speaker modeling for speaker verification and its adaptation to the task of speaker diarization. Then the baseline binary key based speaker diarization system is described in detail. Finally, the implementation of the baseline system is evaluated. Evaluation results obtained give insight about the potential points of the system to be improved.

Chapter 4 presents the main contributions of this thesis to single-session speaker diarization based on binary keys. First, some initial attempts are described and evaluated. Second, an in-depth analysis of the binary key modeling for speaker diarization is conducted. Conclusions extracted from this analysis, as well as the conclusions extracted from the evaluation of the baseline system lead to propose improvements in some modules of the system. These improvements aim at improving both performance and efficiency. Finally, the proposed improvements are validated experimentally.

Chapter 5 proposes a new binary key system for the task of cross-session speaker diarization. The proposal is first described in detail and then evaluated experimentally.

Finally, Chapter 6 summarizes the major contributions and results of this thesis and proposes lines for future work.

Chapter 2

State of the art

This chapter gives an overview of research conducted in the topic of speaker diarization. [Tranter and Reynolds, 2006] and [Anguera et al., 2012b] are excellent papers which provide in-depth reviews of the state of the art in speaker diarization. For this reason, the review provided here is not as complete, but more focused on the particular aspects that this thesis aims to cover. First, the speaker diarization task is introduced. Then, classic approaches which are still widely used are reviewed, including feature extraction, speech activity detection, speaker segmentation, and speaker clustering. After that, more recent research lines are reviewed, including new methods for speaker diarization derived from the last advances in the field of speaker recognition, detection of overlapping speech, attempts towards fast speaker diarization, and the recently proposed cross-session speaker diarization task. Finally, the most used evaluation metrics for speaker diarization are discussed.

2.1 Introduction

Speaker diarization is the task of segmenting an input audio stream into speaker homogeneous segments and grouping them into speaker clusters according to the speaker identities. In other words, speaker diarization consists in answering the question "who spoke when?" in an audio stream. Most of the times speaker diarization is applied in an unsupervised way, i.e. no prior information about the number of speakers nor their identities is available. However, this is not a required condition. Speaker diarization can be applied in situations where there is prior information available. For instance, in telephone conversations, the number of speaker is usually two, and this information can be exploited and taken into account in the design of systems.

Research on speaker diarization started in the early nineties but was usually referred to separately as speaker segmentation and speaker clustering. It was later in the early two-thousands when the term speaker diarization was coined in the ambit of the NIST Rich Transcription (RT) evaluations.

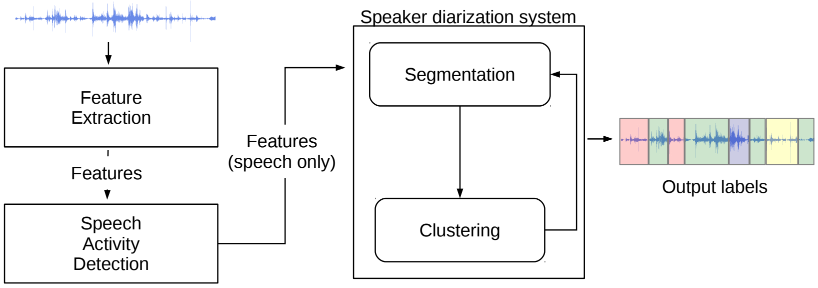


Figure 2.1: Scheme of a generic speaker diarization system

Speaker diarization could also be seen as a particular case of audio diarization where the acoustic classes are the different speakers [Reynolds and Torres-Carrasquillo, 2004].

Figure 2.1 depicts the structure of a generic speaker diarization system. Generally, first of all feature extraction is performed in order to convert the input speech signal into a suitable representation which allows to separate between speakers. After that, a speech activity detection stage is applied in order to filter out the feature vectors corresponding to non-speech content. The resulting speech feature vectors are passed to the diarization system, which performs two different tasks. Segmentation aims at finding speaker change points in the audio stream, providing a set of speaker homogeneous segments. Clustering aims at organizing those segments into speaker groups or cluster according to speaker identities. As it will be seen later, segmentation and clustering may be performed as separate, independent processes or can go hand in hand within an iterative process.

2.2 Speaker diarization review

In this section, a review of the basic approaches to speaker diarization is done, putting the focus on the different components involved in the overall process. Those components include feature extraction, speech activity detection, speaker segmentation, and speaker clustering.

2.2.1 Features for speaker diarization

One of the steps involved in any pattern recognition system is the extraction of features from input data. Features are measures of characteristics which allow the distinction between different data classes. Ideally, those features should provide both high inter-class and low intra-class discrimination at the same time.

As the aim of a speaker diarization system is to segment and to group audio input data into speaker homogenous segments, feature extraction has to provide a representation of the signal which enables the system to optimally separate the participating speakers. Therefore, features for speaker characterization (recognition and identification) seem to be a suitable option for speaker diarization, as both tasks are closely related. Those features for speaker characterization are also widely used for speech recognition. The most popular features include Mel Frequency Cepstral Coefficients (MFCC), Linear Frequency Cepstral Coefficients (LFCC), Perceptual Linear Predictive coding (PLP), Linear Predictive Coding (LPC), and other short-term, low-level features derived from the speech spectrum.

In principle, those parametrization techniques were not designed to discriminate between speakers. Contrarily, they were developed for automatic speech recognition, where phonetic information must be captured, independently on the speaker. However, as those feature extraction methods rely on the human hearing perception (MFCC, PLP), or on the human speech production (LPC), it seems reasonable to think that those features may be useful for identifying individuals, as humans do.

The fact is that those feature extraction methods are extensively used in state-of-the-art speaker-related systems, yielding good performance (usually with a higher number of coefficients, as it has been shown to capture speaker information). However, the community has taken this speech/speaker recognition contradiction into account, which has led to some research towards features specifically designed for characterizing speakers, either used alone or in conjunction with other classic feature extraction approaches.

Following this argument, [Yamaguchi et al., 2005] proposes a series of features extracted from: energy, pitch, peak-frequency centroid, peak-frequency bandwidth, temporal feature stability of the power spectra, spectral shape, and white noise similarities, in order to perform audio segmentation into several classes such as: speech, silence, noise and crosstalk. Also in this line, [Huang and Hansen, 2006] analyzes the Perceptual Minimum Variance Distortionless Response, Smoothed Zero Cross Rate, and Filter-bank Linear Coefficients for the task of speaker segmentation. Although those features outperform classic MFCC features, the improvement is not significant enough to replace them.

In order to reduce channel mismatch effects, feature warping and normalization have also been explored for speaker diarization, gaining relative success according to the domain of application. [Pelecanos and Sridharan, 2001] and [Ouellet et al., 2005] use feature warping with the purpose of reducing channel mismatch effects. These works aim at changing the shape of the distribution of the features to a Gaussian shape. [Sinha et al., 2005] successfully applies feature warping for speaker diarization of broadcast news data, whilst [Zhu et al., 2006] applies it for meeting data. [Nguyen, 2003] proposes a non-linear feature normalization based on the Riemannian differential geometry, which improves performance when used combined with standard features. However, it is not clear how feature warping and/or normalization techniques are always beneficial for speaker diarization, as they may remove useful information for discriminating between speakers. For example, [Kenny et al., 2010] has reported a decrease in performance when using normalized features against un-normalized ones.

Over the years, short-term, low-level features have become standard for speaker-related tasks. However, speech contains information at different levels from which features can be extracted. From the point of view of their physical interpretation, features can be divided into short-term spectral features (20-30 ms), voice source features, spectro-temporal features (tens or hundreds of ms), prosodic features (pitch, energy, duration, rhythm) and high-level features (conversational-level patterns) [Kinnunen and Li, 2010]. Generally, the higher the level of analysis is, the more robust the features are, but the more difficult to extract. Prosodic features have been successfully applied in speaker recognition [Shriberg et al., 2005]. In [Friedland et al., 2009b], prosodic and other long-term features were studied in a speaker diarization system. The authors demonstrate how the use of some prosody-related features combined with standard MFCC improves performance. Once again, this improvement is not significant enough to extend these features widely. Finally, in [Imseng and Friedland, 2010] a cluster initialization method based on prosodic features was presented. This approach showed to be more robust than other initialization methods, as it relies on prosodic features, which are less affected by the environmental conditions than lower-level, short term features.

When speech is recorded with more than one microphone (meeting audio), additional information on the different speech sources can be extracted. This information is reflected in the time-delays of the signal among the microphones. Early work [Lathoud and McCowan, 2003] needed prior knowledge of microphone placement. However, a priori microphone location information is not always available. The first work that does not rely on microphone placements [Ellis and Liu, 2004] showed potential results, although still far from performances of systems using acoustic-based features. In [Ajmera et al., 2004], a first combination of acoustic and time-delay features was presented. The work showed potential, although microphone locations were still needed. In the context of the NIST evaluations, the combination of acoustic and inter-channel delay features at the weighted log-likelihood level was increasingly becoming more popular, but the optimum weights were dependent on the meeting [Pardo et al., 2006], [Pardo et al., 2007]. [Anguera et al., 2007] improves the previous work through automatic weighting based on an entropy-based metric, and also proposed a complete front-end with multiple microphones. More recently, [Evans et al., 2009] presented an approach to unsupervised discriminant analysis of inter-channel delay features. This work reported performance around 20% DER by using only delay features. Today, the use of delay features is widely extended and has become a standard in diarization of meetings recorded with multiple microphones.

Finally, some work has been done in the field of multi-modal diarization by combining audio and video streams. In [Friedland et al., 2009a], MFCC features are combined with compressed domain video features to improve performance of a speaker diarization system.

2.2.2 Speech activity detection

Speech Activity Detection (SAD), also referred to as Voice Activity Detection (VAD), is a fundamental pre-processing task for many speech-related applications, such as coding, enhancement and recognition. SAD consists in labeling an audio input stream into speech and non-speech content. This way, systems can use the produced labels to keep the audio segments of interest, i.e. the speech content, and discard no relevant audio content, i.e. non-speech.

SAD could be considered a particular case of the more general task of audio segmentation in which there are only two audio classes: speech and non-speech. However the non-speech class is usually broken down into subclasses according to the domain of interest. For instance, in a meeting context, non-speech content could involve silence, but also other audible non-speech content, such as door knocks, paper shuffling, breathing, coughing or laughing. When using broadcast TV or radio data, music is commonly present. Even combinations of sounds can occur (speech over background noise or over background music). Therefore, the considered audio classes should be selected according to the nature of the audio being processed.

Accurate SAD is crucial in speaker diarization for two main reasons. First, the Diarization Error Rate (DER) metric, commonly used to evaluate diarization performance, takes into account both false alarm and missed speech error rates, which come directly from the SAD module. Therefore, poor SAD performance will produce a systematic increment of DER. Second, false alarm error rate has an additional impact on the speaker models generated during the speaker diarization task. Having a high false alarm error rate implies that non-speech content is being labeled as speech. Therefore, speaker clusters become impure or "polluted". In consequence, the trained models become less discriminative, introducing additional difficulties in segmentation and clustering steps [Wooters et al., 2004], [Fredouille and Evans, 2007].

One can find many different SAD approaches and studies in the literature [Ramírez et al., 2007], [Khoa, 2012]. However an overview of the SAD techniques most used in speaker diarization is given next.

The first attempts in speaker diarization usually considered SAD as an integrated process within the system by keeping a non-speech cluster. However it was observed that an independent pre-processing SAD system provided better results. Given the heterogeneity of sounds which can be present in any speech recording, with variable energy levels within the non-speech class, simple energy detectors have been shown to be ineffective. This is why model-based approaches have prevailed over energy-based ones. Some systems using GMM-based SAD are described in [Zhu et al., 2008], [Wooters et al., 2004], [Fredouille and Senay, 2006] and [van Leeuwen and Konecný, 2007]. The system presented in [Fredouille and Evans, 2008] makes use of Maximum A Posteriori (MAP) model adaptation in order to fit specific acoustic conditions. Alternatives to GMM have also been proposed, such us Linear Discriminant Analysis (LDA) classifiers [Rentzeperis et al., 2006], or Support Vector Machines (SVM) classifiers [Temko et al., 2007].

Although model-based strategy tends to work better than energy-based, it presents robustness issues against changes in the acoustic conditions, as it relies on external data for training the acoustic classes models. It is for this reason that some approaches combine both energy-based and model-based detection, and are referred to in the literature as hybrid approaches. Usually, a first pass of an energy detector is performed to label a limited amount of speech and non-speech for which high confidence has been estimated. Then, the labeled data is used to obtain specific speech and non-speech models for the input audio. Some works in this direction are described in [Anguera et al., 2006], [Wooters and Huijbregts, 2008] and [Nwe et al., 2009].

More recently, alternative methods to SAD have been proposed. For SAD over telephone conversations, the Hungarian phoneme recognition tool [Schwarz, 2009] has been widely used to infer speech and non-speech regions. [Aronowitz, 2007] proposes a segmental approach to SAD, in which the input signal is divided into segments of 0.03s and represented by Gaussian supervectors. Then, the classes can be modeled by generative models such as GMMs or can be classified by SVMs. [Castán et al., 2011] performs factor analysis after a first HMM/GMM-based audio segmentation pass, in order to classify speech segments into speech and speech plus noise on a broadcast news context. Finally, a Deep Neural Network (DNN) approach to SAD is proposed in [Saon et al., 2013]. This approach outperformed other classic methods over very noisy conditions.

2.2.3 Speaker segmentation

Speaker segmentation aims to detect the points in the audio stream where a speaker change is produced. The result of this process is a set of audio segments, each one containing audio from a single speaker. In general, three different approaches can be found in the literature, depending on the availability of training data. First, the most common situation is that where no training data is available. In such situation, the metric-based approaches are the most suitable. Second, if there is prior information or training data of the participating speakers, a model-based approach can be adopted. Models can be trained on labeled speaker data and a decoding phase can be applied over the audio stream in order to find the speaker homogeneous segments. The third is the silence-based segmentation, which can be used with the assumption that all speaker segments are delimited by silence. The most widely used approach is the metric-based segmentation, as it does not require training data. However, combinations of metric-based and model-based can be performed. First, a metric-based segmentation is applied, and at the end of the process a model-based pass can be applied to refine speaker segment boundaries. In this section, a brief review of metric-based and model-based segmentation is given.

2.2.3.1 Metric-based segmentation

Given an audio stream, any point could be a potential speaker change point. The classical approach performs a hypothesis test on those points. Given a data window of two adjacent (or even overlapped) segments surrounding a potential speaker change point, two hypotheses are evaluated. The first one (H_0) is that both segments comes from the same speaker and, therefore, there is no speaker change. The second (H_1) is that each segment comes from a different speaker, and therefore there is a speaker change point. To perform this test, three different models are trained: one unique model trained on the two segments representing hypothesis H_0, and two models (one per each segment) representing hypothesis H_1. If there is actually a speaker change point, the two model solution will be more accurate than the one model one, and vice versa. Therefore, the problem is reduced to the task of discovering (by means of some metric) which of the two modeling solutions better fits the window being processed. Commonly, this process is applied along the entire audio stream using a sliding window, so it is a computationally costly process.

Among others, the most used metrics are the Bayesian Information Criterion (BIC), the Generalized Likelihood Ratio (GLR), and the Kullback-Leibler divergence.

The Bayesian Information Criterion is probably the most used metric for speaker segmentation. BIC was introduced by [Schwarz, 1978] as a measure of how well a model fits the data. Given a data set X of N samples, and being Θ a model which describe the data, BIC is defined as:

$$BIC(\Theta) = \log(\mathcal{L}(X|\Theta)) - \lambda\frac{1}{2}\#(\Theta)\log(N) \tag{2.1}$$

where $\log(\mathcal{L}(X|\Theta))$ is the log-likelihood of the data given the model, λ is a free parameter dependent on the data, and $\#(\Theta)$ is the number of free parameters to estimate in model Θ. BIC is, therefore, a likelihood criterion penalized by the model complexity.

In order to use BIC to evaluate if a change point occurs between both segments i and j in the window, the BIC increment ΔBIC is defined as:

$$\Delta BIC = BIC(H_1) - BIC(H_0) = R(i,j) - \lambda P \tag{2.2}$$

where $R(i,j)$ is the difference between the log-likelihoods obtained for each hypothesis, and P is the complexity penalty term. When each segment is modeled by a full-covariance Gaussian distribution, $R(i,j)$ can be written as:

$$R(i,j) = \frac{N}{2}\log(|\Sigma_{i,j}| - \frac{N_i}{2}\log(|\Sigma_i|) - \frac{N_j}{2}\log(|\Sigma_j|) \tag{2.3}$$

and the penalty term for a full covariance matrix as

$$P = \frac{1}{2}(p + \frac{1}{2}p(p+1))\log(N) \tag{2.4}$$

When GMMs are used to model the segments, Equation 2.2 is written as

$$\Delta BIC = \log\mathcal{L}(X,\Theta) - (\log\mathcal{L}(X_i,\Theta_i) + \log\mathcal{L}(X_j,\Theta_j)) - \lambda\Delta\#(i,j)\log(N) \tag{2.5}$$

where $\Delta\#(i,j)$ is the difference between the numbers of free parameters in the two modeling options (combined model versus two individual models).

For speaker segmentation, the first work using ΔBIC was [Chen and Gopalakrishnan, 1998], and from then, it has been widely used, becoming the most popular metric.

A disadvantage of ΔBIC is the presence of the free parameter λ, which has to be tuned for the used data. λ was introduced in order to adjust the complexity penalty term. To avoid the use of λ, [Ajmera and Wooters, 2003] uses a model $\Theta_{i,j}$ for H_0 with a number of free parameters equal to the sum of the number of parameters of models Θ_i and Θ_j estimated for H_1.

A second widely extended metric for speaker segmentation is the Generalized Likelihood Ratio (GLR). It defines a ratio between the two hypotheses H_0 (both segments are uttered by the same speaker) and H_1 (the segments are uttered by different speakers). The GLR is calculated as

$$GLR\left(\frac{H_0}{H_1}\right) = \frac{\mathcal{L}(X_{i,j}|\Theta_{i,j})}{\mathcal{L}(X_i|\Theta_i)\mathcal{L}(X_j|\Theta_j)} \tag{2.6}$$

where $\Theta_{i,j}$ is the model trained on the two segments $X_{i,j}$, and Θ_i and Θ_j are the models trained on X_i and X_j respectively, and $\mathcal{L}$ indicates likelihood. For speaker segmentation, the distance between the two hypotheses $D(i,j) = -\log(GLR(\frac{H_0}{H_1}))$ is used to determine if a speaker change point occurs or not, according to a threshold empirically estimated. Some of the first works using GLR for speaker segmentation are [Liu and Kubala, 1999; Delacourt and Wellekens, 2000; Bonastre et al., 2000].

Finally, a third much extended metric for speaker segmentation is the Kullback-Leibler divergence (KL). Given two distributions P and Q, KL divergence measures the expected number of extra bits required to code samples from P using a code based on Q. If both distributions are Gaussian, then the KL divergence is defined as

$$KL(P||Q) = \frac{1}{2}\left(\mathrm{tr}(\Sigma_q^{-1}\Sigma_p) + (\mu_q - \mu_p)^t\Sigma_1^{-1}(\mu_q - \mu_p) - k - \ln\left(\frac{\det\Sigma_p}{\det\Sigma_q}\right)\right) \tag{2.7}$$

KL gives a measure of how different two distributions are, but it is not strictly a distance measure since KL divergence is no symmetric. Instead, a symmetrized version is used, referred to as Symmetric Kullback-Leibler (KL2), and defined as

$$KL2(P||Q) = KL(P||Q) + KL(Q||P) \tag{2.8}$$

KL2 for speaker segmentation was first used in [Siegler et al., 1997] in a broadcast news context.

2.2.3.2 Model-based segmentation

When there is prior knowledge about the speaker/classes and there is training data available, every class/speaker can be modeled on this data, and audio or speaker segmentation may be performed by means of some decoding process by using a close set of class/speaker models. If the speakers are known a priori, specific models for each speaker can be obtained. Otherwise, more general speech classes, such as male/female speech, wide-band or telephone speech can be used to guess the potential speaker change points. Usually, GMMs are used as models and classification is done by maximum likelihood or Viterbi decoding.

Once the whole speaker diarization process has been done, the resulting speaker clusters could be used as training data to obtain speaker models. Then the input data could be segmented again through model-based speaker segmentation. This process is usually referred to in the literature as speaker re-segmentation [Reynolds and Torres-Carrasquillo, 2005], and it is commonly applied at the end of diarization in order to refine speaker boundaries.

2.2.4 Speaker clustering

Speaker clustering aims at grouping all the speech segments that belong to the same speaker into an unknown number of speaker clusters. Normally, speaker clustering is the stage applied just after the speaker segmentation process, with the aim of organizing the resulting speaker homogeneous segments into speaker clusters.

Given an initial set of speech segments, there are a number of different partitions in which they can be distributed. Every hypothetical partition consists of set of non-overlapping clusters containing all the segments. The desired partition is unique and must be selected from all possible partitions. Exploring all possible partitions is normally unfeasible, so suboptimal methods have to be used. The preferred clustering technique in the literature is referred to as Hierarchical Clustering. Hierarchical clustering performs local-optimum-based decisions that increasingly reduce the search space. There are two main approaches to hierarchical clustering, namely the bottom-up and the top-down approaches.

- **Bottom-up**: Also known as Agglomerative Hierarchical Clustering (AHC), this method starts with a number of clusters greater than the actual number of speakers. Usually, the initial clustering is the finest one, in which each segment forms a cluster by itself, or alternatively, the initial clusters are obtained by means of some cluster initialization technique. Then, the closest cluster pair is iteratively merged until a stopping criterion is met. Bottom-up is by far the most used approach to speaker clustering in speaker diarization, directly applied over the segments obtained in the segmentation stage. Some examples of systems using this clustering approach are [Jin et al., 1997; Siegler et al., 1997; Wooters et al., 2004]
- **Top-down**: Also known as divisive approaches, these methods start from a number of clusters smaller than the actual number of speakers (commonly a single cluster containing all the segments together), and those clusters are iteratively split until a stopping criterion is met. This approach is used in works like [Meignier et al., 2001; Fredouille and Senay, 2006]

Both bottom-up and top-down approaches make local decisions in order to decide which cluster pair should be merged, or which cluster should be split, respectively. The use of some kind of distance measure between clusters is needed to compare speaker clusters. The most used distances are those also used for speaker segmentation (BIC, GLR, KL2), being the BIC distance the most popular. In this case, ΔBIC is calculated for all possible cluster pairs at every iteration, and the cluster pair providing the highest ΔBIC value is merged.

The iterative process is performed until a certain stopping criterion is met. The stopping criterion is a key point of the process since it will determine the final number of clusters, which will ideally be equal to the actual number of speakers. The most straightforward stopping criterion is that of using a threshold value in the distance measure of cluster similarity. If none of the cluster pairs fulfill the threshold, then the process is stopped. However, this approach may suffer a lack of robustness when there is mismatch between training and test conditions.

Once again, the most used stopping criterion is ΔBIC. When $\Delta\text{BIC} < 0$ for all cluster pairs, the process finishes. Although the threshold is implicitly fixed, BIC has a free parameter λ for controlling the penalty term that needs to be tuned on development data. As for BIC-based speaker segmentation, the use of the parameter λ can be avoided by compensating the complexity of the models involved to represent the two hypotheses H_0 and H_1, as explained in [Ajmera and Wooters, 2003].

2.2.5 One-step segmentation and clustering

Speaker segmentation and speaker clustering are two core modules of speaker diarization originally proposed as independent processes. Over the years, many systems have taken this approach [Barras et al., 2006; Rouvier et al., 2013].

One of the main limitations of applying both processes separately is the propagation of early errors made in the segmentation step. Particularly, miss-detected speaker turns are not recoverable in the clustering step. To cope with this limitation, a last stage called re-segmentation is added after the clustering stage to recover possible missed speaker turns and refine segment boundaries. In this stage, models (usually GMMs) are trained on the obtained speaker clusters, an HMM is built by adding a state per each speaker, and Viterbi decoding is performed to obtain the final segmentation.

However, other systems perform both segmentation and clustering at the same time. The first system applying this approach within a bottom-up scheme is the one described in [Ajmera and Wooters, 2003], and later the approach was adopted by others [Wooters and Huijbregts, 2008; van Leeuwen and Konecný, 2007]. On the other hand, the one-step segmentation and clustering within a top-down scheme was first introduced in [Meignier et al., 2001]

In this approach, each cluster at the current iteration is modeled by a GMM or HMM trained by Maximum Likelihood (ML) or by adapting a Universal Background Model (UBM) by Maximum a Posteriori (MAP) adaptation. Then, the data is re-assigned to the clusters through Viterbi decoding. Sometimes this process is repeated several times until no variation in the segmentation is observed. After that, the closest cluster pair is merged by calculating pair-wise cluster distances using some of the similarities explained in Section 2.2.3.1. Finally, the model for the new cluster is estimated. This process is iteratively repeated until there is only one cluster remaining or until the stopping criterion is met.

This scheme has the drawback of being slower than if segmentation and clustering are performed separately. However it enables the re-allocation of data across the iterations, with the consequent refinement of speaker segment boundaries.

Usually, duration restrictions are applied on the HMM in order to force a minimum segment duration. This minimum length is set to the considered minimum duration of a speaker turn.

2.3 Current research lines

The previous sections give an overview of the most relevant classic approaches to speaker diarization widely used by the community. This section reviews some of the most recent efforts carried out to overcome some limitations of classic systems. First, some novel frameworks for speaker diarization are described, both designed specifically for the task, and derived from the last advances in the field of speaker recognition. Then, the important challenges of detecting and handling overlapping speech are addressed. Next, efforts done on speeding up speaker diarization are reviewed. Finally, the recently proposed task of cross-session speaker diarization and its main approaches are discussed.

2.3.1 New speaker diarization methods

The vast majority of current state-of-the-art systems are based on the approaches to segmentation, clustering and/or one-step segmentation and clustering described above. However, other techniques have been explored in the literature.

In [Vijayasenan et al., 2009], a novel information-theoretic framework to speaker diarization based on the information bottleneck principle was proposed. This approach aims to minimize the loss of mutual information between successive clusterings while preserving as much information as possible from the original dataset. It provides similar performance to classic GMM-based systems over meeting recordings, but at a lower computational cost.

Another approach gaining attraction is based on Variational Bayes training (VB) [Attias, 2000]. This framework enables to learn the model parameters and adjust complexity of models according to the available training data within the same algorithm. VB was first used for speaker clustering in [Valente and Wellekens, 2004]. More recently, [Kenny, 2008; Kenny et al., 2008] successfully applied it to speaker diarization using eigenvoice modeling.

The last advances in speaker recognition have provided a series of modeling approaches which have been adapted to the speaker diarization task. The GMM supervector modeling provides a compact representation of a speaker utterance. A GMM supervector is the result of concatenating the mean vectors of all components of a GMM trained or adapted on a speaker utterance/cluster [Campbell et al., 2006]. It provides a fixed length representation of the utterance, regardless of its duration, and enables direct comparison between GMM supervectors by just using similarity measures. In speaker diarization, each speaker segment can be converted into a supervector. Then clustering can be performed over the set of supervectors. For instance, in [Aronowitz, 2010], GMM supervectors are computed for each segment and the Nuisance Attribute Projection [Solomonoff et al., 2004] is applied to remove the intra-session intra-speaker variability.

More recently, Joint Factor Analysis (JFA) [Kenny et al., 2008] have gained attention in speaker recognition. JFA aims at compensating the channel- or speaker- variability by exploiting prior knowledge about the speaker space in order to find a low dimensional vector of speaker factors. A first effective factor analysis for speaker diarization was proposed in [Castaldo et al., 2008], and later it was extended in [Kenny et al., 2010].

Following this trend of channel variability compensation, the total variability/i-vector paradigm (nowadays the state-of-the-art in speaker recognition and language recognition) has been successfully applied to speaker diarization of telephone conversations recordings.

In this scenario, where the number of speakers is known a priori (two speakers), K-means algorithm can be effectively applied to cluster the i-vectors representing the segments, as it has been shown in [Shum et al., 2011] and [Dehak et al., 2011]. Similarly, in [Vaquero et al., 2010] K-means is applied to vectors of speaker factors extracted from the segments.

When the number of speakers is not known a priori, AHC is usually used to cluster a set of i-vectors or vectors of speaker-factors. In [Silovsky and Prazak, 2012], a two-stage diarization scheme is proposed, where the first pass performs BIC clustering configured to obtain an under-clustered solution. Then the second pass calculates i-vectors for the set of clusters and performs AHC using the cosine distance as similarity measure. The process stops when the maximum similarity is below a threshold. [Rouvier et al., 2013] follows a similar two-stage scheme, but the second pass can be performed in two different ways. The first applies a Cross Likelihood Ratio (CLR) based clustering. The second transforms the clusters from the first stage to i-vectors and performs a global clustering defined as an optimization problem of Integer Linear Programming (ILP), where the objective function is to minimize the number of clusters and to minimize the dispersion within the clusters.

2.3.2 Overlap detection

One of the main open challenges in speaker diarization is the overlapping speech detection and handling. Currently most of the systems do not deal with overlapping speech, thus in practice only one speaker label is assigned even if more than one speaker are speaking at the same time instant. Overlapping speech is present in many audio domains: meeting recording, telephone calls recordings, TV and radio programs, etc, and can constitute a big part of the total system error.

All the overlapping speech not labeled by the system becomes miss speaker time, which is summed to the overall DER. A second problem of including overlapping speech is that those regions will be clustered into single-speaker clusters, and therefore purity of those clusters will be affected, as noise is being introduced. Even if a diarization system is not intended for assigning more than one label to the overlapping regions, removing such regions from the overall process would help the clustering stage.

In [Boakye et al., 2008], initial work on detecting overlapping speech was done, proposing features such as MFCC, root-mean-squared energy and Linear Predictive Coding (LPC) residual energy, getting slight improvements in diarization performance. [Otterson and Ostendorf, 2007] presents a theoretical study of overlapping speech using ground-truth labels. In this work, diarization performance is increased by assigning a second speaker on overlapped regions deduced from the labels of the neighbor segments, as well as excluding overlapping speech from the input of the diarization system. Today combinations of MFCC and other features such as modulation spectrogram features or prosodic features are usually used [Hernando and Zelenák, 2012]. More recently, a different approach to overlapping speech detection based on long-term conversational features was presented in [Yella and Bourlard, 2014]. This work proposes to improve a short-term spectral feature based overlap detector by incorporating information from long-term conversational features in the form of speaker change statistics, calculated over segments of a few seconds. This approach is motivated by the observation that segments containing more speaker changes are more likely to present more overlapping speech.

2.3.3 Speeding up speaker diarization

As it has been discussed, most state-of-the-art systems rely on the use of Gaussian Mixture Model (GMM) for speaker segmentation and clustering, trained on acoustic features using maximum likelihood. In speaker segmentation, GMMs and Bayesian Information Criterion (BIC) are intensively computed within a sliding window in order to decide whether there is a speaker change point or not. In speaker clustering, speaker clusters are also usually modeled by GMMs, and compared using BIC to decide which cluster pairs should be merged. Next, BIC is also widely extended as a stopping criterion. Moreover, a final re-segmentation stage may be carried out through Viterbi decoding to refine the output clustering.

In conclusion, all the mentioned algorithms are applied iteratively, imposing a high computational load which results in processing times which may be too long for some real-life applications (above 1xRT, being xRT the Real Time factor).

Today, given the great increase of audiovisual content, algorithms able to process large quantities of data in a reasonable time period are strongly desirable, and increasingly required. In this line, some efforts have been done in order to get faster systems. In [Huang et al., 2007], fast-mach methods are investigated for reducing the hypothesis space of the BIC approach and selecting the most likely clusters to merge, with the consequent computational savings, getting 0.88 xRT. Although faster than real-time, this approach seems not fast enough to process large databases quickly. The framework proposed in [Vijayasenan et al., 2009] already discussed in Section 2.3.1 not only obtained performance close to other state-of-the-art systems, but also great computation savings, achieving around 0.3 xRT. Further on, the use of parallel hardware was investigated. [Gonina et al., 2011] proposes a parallel implementation of the GMM training for a GPU, achieving great speed ups between 0.02-0.004 xRT. However, the approach is very dependent on complex, non-standard hardware architectures and low-level programming methodologies.

Later, in [Anguera and Bonastre, 2011] a novel speaker diarization system based on the “binary key” speaker modeling was presented. The binary key speaker modeling was introduced in [Anguera and Bonastre, 2010] for speaker verification. This approach performs a conversion of the feature vectors into a binary representation called binary key. A binary key is a vector of zeros and ones which is able to preserve speaker related information contained in the source sequence of feature vectors. This conversion is done thanks to a UBM-like model called Binary Key Background Models (KBM) trained on a database of anchor speakers. Intuitively, a position i of a binary key set to 1 indicates that the i-th Gaussian component of the KBM coexists in the same acoustic area as the data being modeled. In other words, the positions of the binary key set to 1 indicate that those Gaussian components in the KBM are the ones which best describe the data. Therefore, a binary key provides a compact representation of an utterance or even a set of utterances for a given speaker in a single binary vector, and can act as a speaker model. In addition, given the binary nature of the representation, binary metrics can be efficiently applied for comparing binary keys. For speaker diarization, a novel method for training the KBM without the need of external training data was proposed in [Anguera and Bonastre, 2011], in which the KBM is trained on the test data itself. After converting the input acoustic feature vectors (MFCCs) to binary keys, an agglomerative architecture is applied. Unlike classical GMM-based systems, all operations are performed in the binary domain, which makes the process very fast. In this work, a xRT value of 0.1 was reported with a slight decrease in performance on all the NIST RT databases of meeting recordings. Later in [Delgado et al., 2014b], the work was extended and applied to the broadcast TV data domain, obtaining similar results both in performance and execution time.

Lately, with the expansion of the last achievements in speaker recognition (Gaussian supervectors, joint factor analysis, i-vectors), new advanced speaker diarization techniques have emerged. Normally, a speech segment/cluster is represented by a single vector of a certain dimension. Once speaker segment/clusters are converted into multidimensional points, speaker clustering is reduced to the task of clustering single vectors. Those vectors can be efficiently compared through some fast similarity measure, such as the dot-score or the cosine distance. Therefore, the recent "supervector" paradigm is presented as a potential fast approach to speaker clustering. In spite of this, speaker segmentation is still commonly faced through BIC-based and other similar approaches. Consequently, a speaker diarization system composed of classic segmentation plus a supervector approach will presumably suffer of computational issues. An example of such an approach is the one described in [Silovsky et al., 2012]. This system first performs speaker segmentation based on a test statistic derived upon the maximum likelihood approach, followed by a two-stage clustering scheme, the first one based on BIC, and the second one based on i-vectors. The acoustic-based segmentation stage shows a 0.14 xRT, although this figure was further reduced to 0.02 by using the output of an ASR system (which enables a drastic reduction of potential speaker change points). With regard to the clustering stage, a real-time factor of 0.05 xRT was reported. Therefore, total 0.19 xRT for the only-acoustic-based system was reported, whilst 0.07 xRT for the ASR-aided system. Even if a great speed up is achieved, an ASR output is required, not to mention the need of enough training data to estimate Universal Background Models (UBM), total variability matrices for i-vector estimations, and other required resources (such as LDA matrices for intra-session variability compensation).

2.3.4 Cross-session speaker diarization

Cross-session speaker diarization aims at expanding the speaker diarization task to a broader context, where speakers participating in different recordings along a collection of audio files must receive the same speaker identifier. These recurrent speakers are called "cross-session speakers". In the literature this problem is commonly referred to as "cross-show speaker diarization" [Yang et al., 2011; Tran et al., 2011]. The term "cross-show" is normally used in works which use TV broadcast data on its experiments. However, in this thesis we use the broader term "cross-session speaker diarization" because the task can be extended to a wide variety of audio documents beyond TV shows, such as meeting recordings, telephone call recordings, and so on. The task has also been referred to as "speaker linking" [van Leeuwen, 2010; Ferras and Boudard, 2012] and as "Speaker attribution" [Ghaemmaghami et al., 2011], but all the three terms refer essentially to the same idea.

Three main schemes for cross-session speaker diarization have been proposed in the literature [Yang et al., 2011; Tran et al., 2011]. These are (1) the global approach by concatenation, (2) the hybrid approach, and (3) the incremental approach.

The global approach by concatenation is the most straightforward and naive method, in which all the sessions in the dataset are concatenated and diarization is performed on the resulted pooled audio file. However, memory and computation requirements of this approach grow significantly with an increasing number of sessions.

The hybrid approach partially solves the computational limitations of the global approach by concatenation by first performing individual speaker diarization on each session. Second, speaker clusters returned by the individual processes are pooled together and a global speaker clustering is performed. This approach is more efficient because the global clustering is done over a limited number of speaker clusters.

Finally, the incremental approach aims to solve some inherent problems of the two previous strategies which make them unusable in a real application scenario. Even the hybrid approach may exceed memory requirements when processing big quantities of data. A second drawback of the hybrid approach is present when the database is increased over time (new sessions are added periodically): in order to perform cross diarization on the new session, the whole global speaker clustering has to be computed again for the complete dataset. To cope with these issues, the incremental approach proposes the use of the information of the sessions already processed to help the diarization of the new session. However, this strategy usually requires an open-set speaker ID system. If new speakers are detected on the new session, then the speaker database is updated.

With regard to the global speaker clustering, BIC [Yang et al., 2011] and Cross Likelihood Ratio (CLR) [Tran et al., 2011] have been used as merging criterion within a global AHC stage. Other approaches use advanced methods like JFA and i-vectors to represent speaker clusters. For example, [Ferras and Boudard, 2012] performs JFA to extract speaker factor posterior distributions which model the output clusters of the single-session diarization system. Then AHC is applied by using some similarity measure to decide cluster merges (cosine distance, symmetric Kullback-Leibler divergence, and Hotelling T-square statistics). The system described in [Dupuy et al., 2012] performs total variability analysis to extract an i-vector for each speaker cluster from the single diarizations. Next, clustering is done by following one of these methods: (1) AHC using Normalized Cross Likelihood Ratio (NCLR), or (2) applying a global clustering formulated as an optimization process of Integer Linear Programming (ILP).

2.4 Speaker diarization evaluation

In this section, the most used metrics for evaluating speaker diarization performance are commented.

Speaker diarization consists of two different tasks: speaker segmentation and speaker clustering. As separated processes, evaluation used to be performed separately as well.

The first metrics for speaker segmentation focused on measuring the missed and false alarm speaker change points. However, this metrics do not take into account the temporal precision of those speaker change points. A second problem of these metrics is that all speaker boundaries are given the same importance when it should not have not to be necessarily this way. For example, a missed speaker change point cannot be recovered later in the clustering stage of a diarization system. On the other hand, a false positive speaker change point is possibly dividing a speaker segment into two smaller speaker segments containing speech from the same speaker, so that they could be merged together again after clustering. In order to avoid the intrinsic limitations of these metrics, time-aligning-based metrics were proposed, where the aim was to measure the amount of time incorrectly assigned to each class. But the problem of this metrics is that the number of classes must be known a priori.

With regard to speaker clustering evaluation, the most extended metrics are the cluster purity and the speaker purity. Given a cluster containing speech segments, its cluster purity is the fraction of the frames corresponding to the most frequent speaker within the cluster and the total number of frames of the cluster. Ideally, all the frames in the cluster should belong to the same speaker, providing a purity value of 1. Given a clustering solution composed of a set of clusters, the total purity of the clustering solution is the time-weighted average of purities of all clusters. This metric indicates if for each cluster there is a dominant speaker or not, but it is not able to provide an actual evaluation of the clustering as a whole. For example, clusters may be very pure, but speakers can be fragmented into more than one cluster. Even each segment can constitute a cluster by itself, so cluster purity would be maximum, but that is not the desired solution.

In this regard, the speaker purity is defined similarly as the cluster purity. But here, the aim is to provide an indicator of how fragmented each speaker is along the set of clusters. As cluster purity, it neither becomes an indicator of quality of the clustering itself. For instance, a single cluster containing all speech segments for all clusters would return the maximum speaker purity.

However, the pair of cluster purity and speaker purity metrics can be use as an indicator to measure clustering performance. The desired partition would be the one which maximizes both metrics. Actually, cluster purity and speaker purity go hand in hand, and normally increasing one of them produces a decrease in the other and vice versa.

The metrics described above evaluate segmentation and clustering separately. Nonetheless, the most popular metric for assessing performance of speaker diarization is the Diarization Error Rate. This metric was proposed by the NIST in the 2000 Speaker Recognition Evaluation [NIST, 2000] for the then-new task of speaker segmentation. This metric take into account both segmentation and clustering errors, as well as errors from the SAD stage. DER is defined as

$$DER = E_{Spk} + E_{FA} + E_{Miss} + E_{OV} \tag{2.9}$$

As it can be seen, DER is the sum of a series of sources of error. E_{Spk} refers to speaker error, defined as the time assigned to incorrect speakers, divided by the total time. E_{FA} refers to false alarm speech, defined as the amount of time incorrectly detected as speech, divided by the total time. This error comes from the speech/non-speech detection. E_{Miss} refers to miss speech, defined as the amount of speech time that has not been detected as such, divided by the total time. This error also comes from the SAD module. Finally, E_{OV} refers to overlapped speech error, which constitutes the errors produced in the detection of overlapping speech when more than one speaker is present in a given segment. These errors are added to miss speech or false alarm errors. On one hand, if the system detects fewer speakers than the actual number of speakers in a segment, the error for the miss-detected speakers is summed to the miss speech error. On the other hand, when the system hypothesizes a number of speakers greater than the actual number of speakers, the error of the extra detected speakers is summed to the false alarm error.

Calculation of DER requires a set of reference segments containing temporal information (beginning and end) plus speaker identity information. As speaker diarization systems assign abstract labels to the detected speakers, tools for calculating DER first has to perform an optimum mapping between system speaker labels and reference speaker labels.

DER is the metric used in many speaker diarization evaluation campaigns such as NIST Rich Transcription, ETAPE, Albayzin, and REPERE evaluation campaigns. First, the primary metric used to be the DER calculated on the regions in which only one speaker is active (NIST RT04 and RT05 campaigns). However, nowadays the standard metric includes overlapping speech in the computation of DER. This was done mainly to encourage authors to study the phenomenon of overlapping speech since, depending on the nature of the audio being processed, it can constitute a big part of the total DER of current systems.

2.4.1 Calculation of DER

As it has been said in Section 2.4, the Diarization Error Rate is the most extended metric to assess speaker diarization performance. Under the ambit of the NIST Rich Transcription Evaluation, evaluation software tools were developed and distributed to the participants, and also released freely on the Internet. These have become the go-to evaluation tools in the speaker diarization community.

Computing the DER involves two steps. The first one consists in establishing a mapping between the hypothesized speakers and the reference speakers. The second step calculates the error rate for the given mapping. As it was explained in Section 2.4, DER is the sum of three kinds of errors: the miss rate, the false alarm rate, and the speaker error rate. It was also pointed out that the miss and false alarm errors also include errors coming from overlapping speech in the case of miss-detecting and over-detecting speakers in such regions, respectively. Finally, the time in error of the three sources of errors is summed, providing the total time in error. In order to obtain a relative measure which could be compared among evaluations with different audio lengths, the total error is normalized according to the total speaker time in the reference. DER is computed for all possible speaker mappings between system and reference labels, and finally the mapping which minimizes the DER is selected as the output mapping.

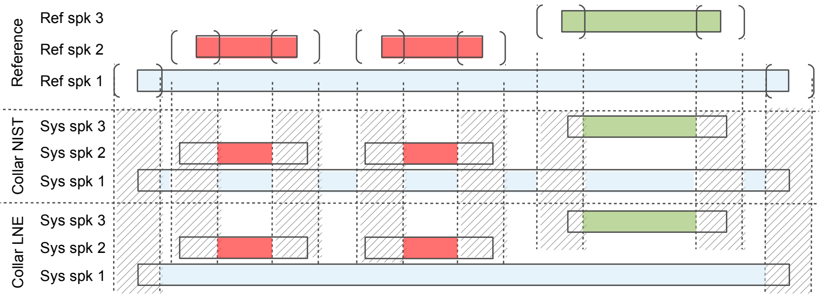


Figure 2.2: Effects of the application of the forgiveness collar on overlapping speech, according to the NIST (middle) and LNE (bottom) methods.

Finally, the last point taken into account by the tools is the intrinsic human imprecision in the elaboration of the references. Defining the exact point where a speech segment starts/ends is really challenging, especially under conditions of noise and overlapping speech. To avoid this limitation, some flexibility has to be added around the segment boundaries. The NIST solution to that problem was the addition of a *forgiveness collar* of +/-250ms around every reference speech boundary. These regions surrounding the frontiers of all reference speech segments are excluded from the evaluation.

Although widely used by the community, it has been found that the NIST tools are not totally adequate in some specific conditions. For example, the NIST tools were not designed to evaluate the increasingly important task of cross-session speaker diarization. A second observation is that the NIST tools tend to discard much overlapping speech at certain situations, due to the application of the forgiveness collar at each speech segment boundary.

Within the Quaero [Yang et al., 2011] and Etape [Gravier et al., 2012] evaluation campaigns, some efforts were done in order to generalize the DER metric in order to fit the needs of such challenging situations. The result of these efforts was a new evaluation tool, developed by the *Laboratory National De Métrologie Et D'essais* (LNE), which introduces some modifications in the calculation of DER. This new methodology is described in [Galibert, 2013]. Essentially, these modifications involve the way the forgiveness collar is applied. Concretely, the forgiveness collar is only applied to the boundaries of the reference speakers in the regions where its system-mapped speaker occurs. This method has one main advantage: In overlapping regions, the collar around the reference boundaries of the reference speaker is only applied to its associated speaker, and it only avoids the computation of miss and false alarm. If in that region there are more speakers present, the collar is not applied to their segments. Figure 2.2 depicts the effect of applying the collar by following the NIST method ("Collar NIST", middle part of figure), and by applying the proposed method ("Collar LNE", bottom part of figure) to a hypothesized system output. In this example, the reference contains a long speech segment uttered by speaker 1. During this turn, two interventions of speaker 2 and one intervention of speaker 3 are produced. It can be observed how the NIST collars for segments of speaker 2 and 3 have an important impact on the segment from speaker 1, provoking an important lost of speech (shadowed regions of the figure). However, with

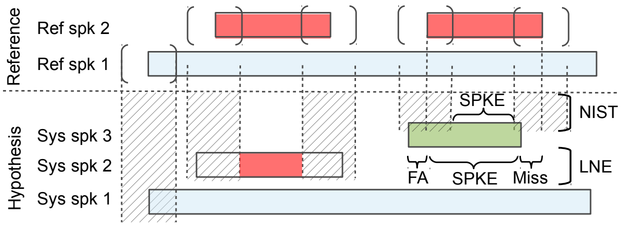


FIGURE 2.3: Collateral effects of the LNE method in regions of speaker error, compared to the NIST method.

the new method, since the collar is only applied to the associated speakers, the speech segment of speaker 1 remains unaltered.

However, it is important to remark that this method has additional consequences. For example, the collar has no effect in regions where the hypothesized speaker is not the one mapped to the reference speaker in such regions. This fact contributes to emphasize all the three kinds of error (speaker, miss and false alarm), as it can be seen in the example depicted in Figure 2.3. In such example, the NIST method applied the forgiveness collar to the segment in error in the system output, and the rest of the segment is accounted as speaker error, while the LNE method does not apply the collar at all, and therefore the whole segment is accounted as time in error, involving speaker error, and false alarm and miss errors when applicable. These phenomena may lead to difficulties to interpret these kinds of errors. In the methodology used by NIST, false alarm and miss errors are exclusively due to the SAD module and the misdetection of overlapping speech. However, with the LNE method, part of those errors also come from the application of the forgiveness collar to the segmentations generated by the diarization system. As a result, getting a meaningful interpretation of false alarm and miss errors can be specially challenging since it is not clear how to determine which part of the error come from the SAD module and which part comes from the application of the forgiveness collar. Nevertheless, throughout this thesis we have chosen this evaluation method since it was the one used in the REPERE evaluation campaign, and therefore its use enable us to compare performance with the official evaluation results.

Chapter 3

Speaker diarization using binary keys

In Chapter 1, the need of fast speaker diarization systems has been argued and emphasized. One of the main objectives of this thesis is to get a very fast speaker diarization system suitable for processing large collections of audio. To achieve that, it is has been decided in this thesis to consider the promising speaker diarization approach based on binary key speaker modeling. The main motivation which supports this decision is its low computational cost, which makes the system very fast. However, the reported results in the literature were still preliminary, and the obtained performance was still a bit far from state-of-the-art systems. Nevertheless, we strongly think that this approach is not mature yet, and that there is still room for further improvement, what leaves a number of open challenges which need investigation.

In this Chapter, first the binary key speaker modeling as it was intended for speaker recognition is studied. Second, a speaker diarization framework based on this speaker modeling technique is provided, putting a special focus on how the binary key speaker modeling is adapted to the different task (although closely related) of speaker diarization. Third, the proposed approach is evaluated in order to establish a baseline performance to be used as a reference for evaluating the proposed improvements developed in this thesis. In addition, the databases used throughout this thesis are described and analyzed. Finally, conclusions are extracted from the obtained results. The in-depth inspection of results sheds light on the main drawbacks of the system and allows identifying the weak points that need to be improved.

3.1 Binary key speaker modeling

The binary key speaker modeling was introduced in [Anguera and Bonastre, 2010] for a speaker verification task. This approach projects a speaker utterance into a binary space which provides a representation in form of a vector of binary values. Its main properties are its small size compared to other speaker modeling techniques, and its low computational cost when comparing different speakers. In addition, the binary key extraction process does not require any hard threshold, which makes the system easy to adapt to different environmental conditions. It also provides a well defined binary domain in which scores and decision are easy to interpret and implement. In fact, comparing speakers just involve the computation of similarity metrics between two binary vectors.

Although this approach was first designed for speaker recognition, it was later successfully applied to speaker diarization [Anguera and Bonastre, 2011] and emotion recognition [Anguera et al., 2012b]. Since the underlying framework is the well-known GMM/UBM paradigm, it is reasonable to think that the binary key modeling could be applied to other audio classification tasks usually addressed by this framework.

The process of conversion of an utterance to a single binary key can be split into two different stages. First, a UBM-like acoustic model called binary Key Background Model (KBM), which constitutes the core of the method, has to be trained. Second, this KBM is used to transform the sequence of acoustic features into a multidimensional vector of binary values called Binary Key (BK).

3.1.1 Binary Key Background Model

The KBM is a UBM-like acoustic model which is used to convert acoustic features into binary features. Such background model makes use of the abilities of the GMM/UBM paradigm to model an overall acoustic space. However, the KBM training process aims not only at capturing the general acoustic space, but also at emphasizing the discriminant aspects in the acoustic space where the speaker specific data is expected to be modeled. In other words, the KBM models the overall acoustic space, but also the speaker specificities. To achieve this goal, a classic UBM is trained to model the overall space, and later extended with a set of anchor speakers to emphasize speaker specificities. The concept of anchor speaker was used before in [Merlin et al., 1999; Mami and Charlet, 2003], and it consists in defining a base of speaker characteristics in which an utterance can be projected.

The KBM training process first involves the training of a classic UBM. Second for each anchor speaker a new GMM is derived from the UBM by Expectation Maximization (EM) or by Maximum a Posteriori (MAP) adaptation. Finally, the KBM is the result of pooling all the obtained GMMs. The number of resulting mixtures is the number of Gaussian mixtures in the original UBM multiplied by the number of anchor speakers.

Later, other methods for estimating the KBM were proposed. In [Bonastre et al., 2011], a more complex scheme is applied in order to obtain a KBM of two levels. The first one consists of a classical UBM. Then, as in the original approach, a GMM is trained for each specific speaker. Next, the set of the i-th Gaussian components of all speaker-specific GMMs constitutes a set of specificities for the i-th Gaussian of the UBM. Finally, a specificity selection process is performed to reduce the final number of specificities. Similar schemes have been used in [Bousquet and Bonastre, 2012; Hernández-Sierra et al., 2012].

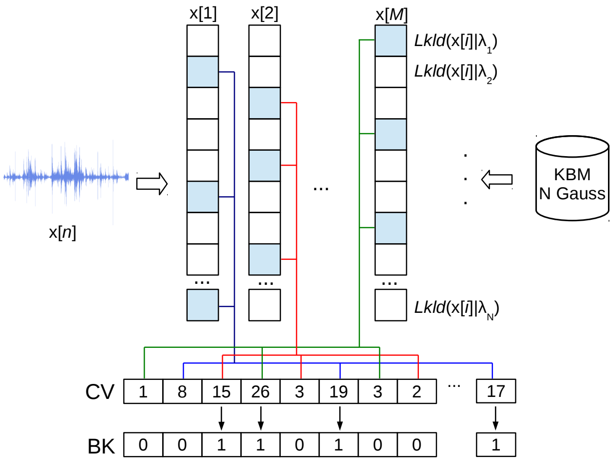


FIGURE 3.1: Binary Key extraction process example.

3.1.2 Binary key computation

Once the KBM is trained, a sequence of input feature vectors can be converted into a Binary Key (BK). Figure 3.1 illustrates the process. A BK $\mathrm{v}_f = \{v_f[1], ..., v_f[N]\}, v_f[i] = \{0, 1\}$ is a binary vector whose dimension N is the number of components in the KBM. The i-th position of v_f represents the i-th Gaussian of the KBM. Concretely, setting a position $v_f[i]$ to 1 (TRUE) indicates that the i-th Gaussian of the KBM coexists in the same area of the acoustic space as the acoustic data being modeled. The BK can be obtained in two steps. First, for each feature vector, the indices of the best N_G matching Gaussians in the KBM (i.e., the N_G Gaussians which provide the highest likelihoods for the current feature vector) are selected. Given a Cumulative Vector (CV) $\mathrm{v}_c = \{v_c[1], ..., v_c[N]\}, v_c[i] \in \mathbb{N}$ (whose positions are initialized to 0), in which each position i also represents the i-th Gaussian in the KBM, those top positions are incremented by 1. Once all the feature vectors have been processed, the CV contains the counts of how many times each Gaussian i in the KBM has been selected for the whole sequence of feature vectors. In other words, the CV stores the relative importance of each Gaussian i in the modeling of the feature vectors set. Intuitively, the procedure projects the acoustic location of each acoustic feature from the feature space to the space of KBM Gaussians, keeping only the components with highest impact. Then, the final binary vector v_f is easily obtained from vector v_c by finding the indices of the top M positions in v_c and then setting those positions to 1 (TRUE) in v_f, and 0 (FALSE) otherwise. This procedure requires the computation of likelihoods of every feature vector given every component in the KBM, and the partial sorting of those likelihoods. This can be implemented efficiently, resulting in a quite fast process.

This process can be seen as a component selection based on highest occupancies. However, the likelihoods are used only to select components on a per frame basis. Thanks to these binary decisions, the likelihood numerical domain is projected to a binary domain.

It is important to remark that parameters N_G and M are not decision thresholds, but parametrization metaparameters used to decide the amount of information extracted for each frame and for the whole feature set, respectively (similar to the role of dimensionality selection of feature vectors).

Finally, note that this method can be applied to any set of features, either a sequence of features from a short speech utterance, or a feature set corresponding to a whole speaker cluster. As it will be shown later, this feature will make the comparison either between utterance-utterance or utterance-cluster pairs straightforward.

3.1.3 Similarity metrics for binary keys

Once the data is in form of binary keys, meaningful similarity metrics are needed in order to perform classification tasks. Given the nature of the BKs, any similarity metric for binary data from the field of information theory could be considered. Some of the available similarity/distance measures are Jaccard, Ghosh, Sokal & Michener, Sokal & Sneath and Yule criteria. In [Anguera and Bonastre, 2010], a similarity metric (based on the Sokal & Michener criterion) between two BKs v_{f1} and v_{f2} is defined as

$$S(\mathrm{v}_{f1}, \mathrm{v}_{f2}) = \frac{1}{N} \sum_{i=1}^{N} (v_{f1}[i] \wedge v_{f2}[i]) \tag{3.1}$$

where $\wedge$ is the bit-wise AND operator. This metric gives the ratio between the number of positions commonly set to 1 in both BKs and the total number of bits N. Later, in [Anguera and Bonastre, 2011], the Jaccard similarity is used, defined as

$$S(\mathrm{v}_{f1}, \mathrm{v}_{f2})) = \frac{\sum_{i=1}^{N} (v_{f1}[i] \wedge v_{f2}[i])}{\sum_{i=1}^{N} (v_{f1}[i] \vee v_{f2}[i])} \tag{3.2}$$

where $\vee$ indicates the bit-wise OR operator. This equation defines the ratio between the number of bits commonly set to 1 and the number of bits not commonly set to 0.

3.2 Speaker diarization based on binary keys

This section describes the binary key based speaker diarization system taken as a baseline in this thesis work, including performance figures. The system is the one described in [Delgado et al., 2014b] (with some minor changes and bug fixes), which is based on the system described in [Anguera and Bonastre, 2011].

The overall system architecture is shown in Figure 3.2. Two well-differentiated modules can be distinguished. First, the acoustic module aims at transforming the input acoustic data into a sequence of Binary Keys. Second, the binary processing module performs a bottom-up Agglomerative Hierarchical Clustering (AHC), following a similar scheme as that of ICSI's system [Ajmera and Wooters, 2003], in which data re-assignment and cluster merges are performed iteratively (there is not a separated speaker segmentation stage), but in this case the binary representation obtained in the previous stage is

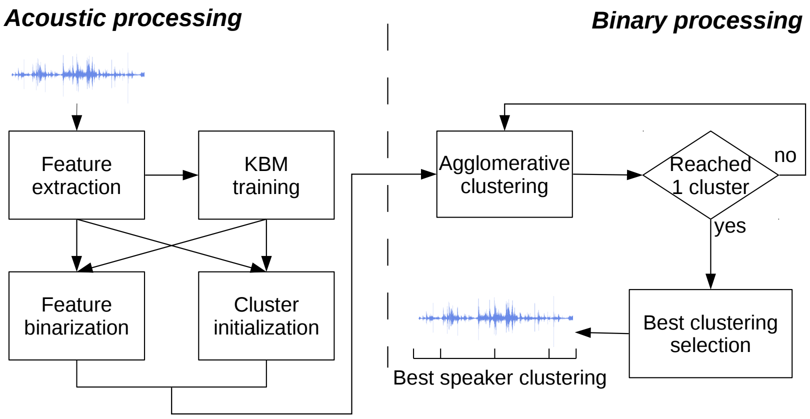


FIGURE 3.2: Overview of the speaker diarization system

used. As it will be shown below, computation load is considerably decreased thanks to the use of the binary representation, compared to a classic GMM-based bottom-up AHC system. Next, the two modules are described in more detail.

3.2.1 Acoustic processing module

The acoustic module performs a transformation of the input acoustic feature vectors into Binary Keys. As it has been shown in Section 3.1, the key element for this transformation is a UBM-like acoustic model, called binary Key Background Model (KBM). The training procedure explained in Section 3.1.1 required the use of a training dataset. In this thesis, an alternative training method which no requires external training data is used. Once the KBM is estimated, the input sequence of feature vectors is converted into a sequence of BKs.

3.2.1.1 KBM training for speaker diarization

The KBM is a UBM-like acoustic model which is used to convert acoustic features into binary features. The original KBM training procedure for binary speaker modeling [Anguera and Bonastre, 2010] requires an external training dataset in order to train an initial UBM which is later adapted to obtain GMMs for the anchor speakers. Although this approach could be applied to speaker diarization, it is reasonable to think that, in a speaker diarization task where no prior information about speaker identities is provided, accuracy will suffer a decrease due to the mismatch between training and testing data. Furthermore, we are after a system that does not require external training data.

Instead, a novel KBM training method for speaker diarization was introduced in [Anguera and Bonastre, 2011]. This method does not require external data: the test data itself is directly used for training.

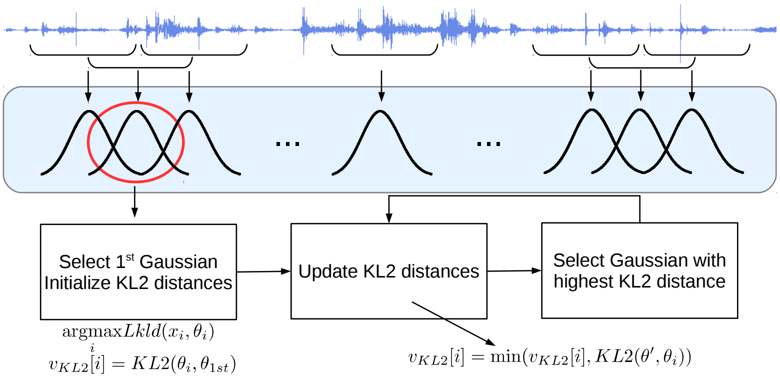

...
...
Select 1st Gaussian
Initialize KL2 distances
Update KL2 distances
Select Gaussian with
highest KL2 distance
$\underset{i}{\text{argmax}} Lkld(x_i, \theta_i)$
$v_{KL2}[i] = KL2(\theta_i, \theta_{1st})$
$v_{KL2}[i] = \min(v_{KL2}[i], KL2(\theta', \theta_i))$

Figure 3.3: Iterative Gaussian component selection algorithm.

The KBM training algorithm for speaker diarization is illustrated in Figure 3.3. First, a pool of single Gaussians is trained on the input features, followed by an iterative process of Gaussian selection which aims to select the most complementary and discriminant components, with the goal of retaining full coverage of the speaker acoustic space of the test data. For Gaussian components estimation, a fixed-length window is used to train single Gaussians, with some window shift (and overlap). The shift value is dependent on the length of the data, and its value is set in order to obtain several hundreds of components. Once the Gaussian pool is obtained, components are selected iteratively until having the desired number of Gaussians. The first selected component is the one which best models the data segment it was trained from (i.e. $\arg\max_{i} Lkld(s_i, \theta_i)$, where θ_i is the Gaussian trained with the i-th segment). For the iterative Gaussian selection, a global dissimilarity vector v_{KL2} is defined to represent distances between the already selected Gaussians to all others remaining in the pool. This vector is first initialized to ∞ as no component is selected yet. The process then works as follows: (1) Compute the KL2 (symmetric Kullback-Leibler) divergence between the previously selected Gaussian θ' and the rest of Gaussians θ_k still not selected, and set $v_{KL2}[j] = \min(v_{KL2}[j], S_{KL2}(\theta', \theta_j))$; (2) Add to the KBM the Gaussian θ^k with the highest dissimilarity with those already selected (i.e. $\arg\max_{k}(v_{KL2}[k])$); (3) Go back to (1) until the desired number of components in the KBM, N, is reached.

It could be argued that a classical GMM trained on the test data could be used instead of the KBM. However, it has been shown in [Anguera and Bonastre, 2010] and [Anguera and Bonastre, 2011] that the KBM is able of produce much more discriminative BKs than a classic GMM. A possible explanation to this fact is that the components of a GMM trained, for example, through iterative Gaussian splitting, model the average acoustic space, whilst the KBM components retain acoustic information of the particular speakers, since most of the components are trained on pure speaker data. A second advantage is its lower computational cost compared to the expectation-maximization algorithm.

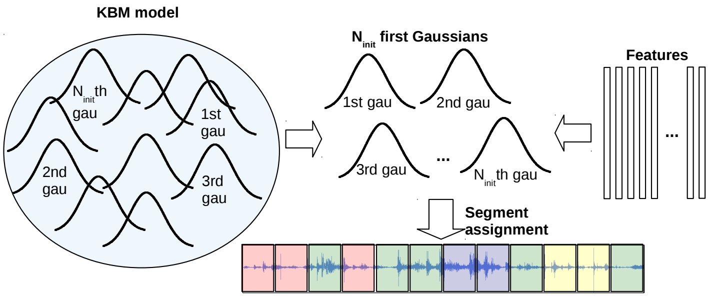


Figure 3.4: Clustering initialization scheme.

3.2.1.2 Converting the input data into binary keys

The aim of the acoustic processing block is to obtain a binary representation of the input data. In this stage, the input data is first divided into short equal-sized segments (with some overlap). Then, each segment is converted into a single BK following the method for BK estimation described in Section 3.1.2 (later, in the binary processing block, each speaker cluster will also be converted into single BKs by using the same method). Once all the data is converted into BKs, comparison between elements is reduced to the application of similarity measures between binary vectors. This fact provides a simple and compact framework in which comparisons between speech-cluster pairs and cluster-cluster pairs are carried out through the same method.

It is important to remark that this process requires the extensive computation of likelihoods of all acoustic features against all the Gaussian components in the KBM. However, these likelihoods are computed only once and can be stored in a table and re-utilized in later steps. The top N_G Gaussians per frame can also be stored and re-used in the computation of new BKs in later stages (cluster BKs computation).

From this point on, all the subsequent processes will be performed over this binary representation, what will result in important computational savings since all operations will be performed on vectors of zeros and ones, involving similarities between binary vectors based on bit-wise, Boolean operations.

3.2.1.3 Clustering initialization

Before switching to the binary processing block, the initial clusters have to be initialized. Clustering initialization in speaker diarization has been extensively addressed in the literature, but the problem is not still solved totally. The original method for cluster initialization for binary key speaker diarization described in [Anguera and Bonastre, 2011] (depicted in Figure 3.4) takes advantage of the first N_{init} components in the KBM as seed models to obtain a first, maximum-likelihood-based, over-segmented clustering of N_{init} clusters. By taking the N_{init} first components, it is thought that the most globally dissimilar Gaussians are selected and then the obtained clustering will be a reasonable starting point. In addition, this method re-uses the likelihoods computed at the beginning

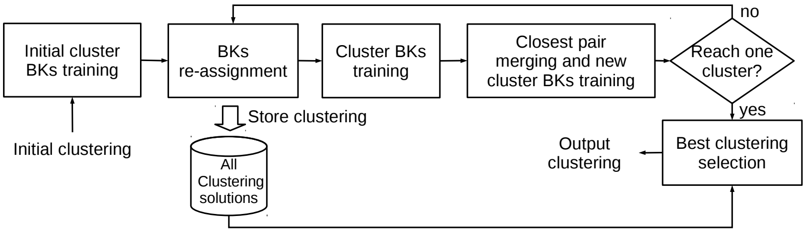


FIGURE 3.5: Scheme of the agglomerative clustering stage.

when obtaining the BKs, so the initialization is efficient. Let us also remark that this is the last step performed in the acoustic domain.

3.2.2 Binary processing module

The binary processing block implements an agglomerative clustering approach. However, all operations are done with binary data, which makes the process faster than in classic GMM-based approaches. The overall procedure is depicted in Figure 3.5. The agglomerative process starts with N_{init} clusters. At each iteration, the input BKs are re-assigned to the current clusters, and the closest cluster pair is merged, reducing the total number of clusters by one. The process is repeated, and after performing N_{init} iterations, the resulting clustering solution will consists of a single cluster containing all speech segments. Finally, the output clustering solution is selected from those intermediate clustering solutions obtained along the whole process.

3.2.2.1 Agglomerative clustering

After estimating BKs for the N_{init} initial clusters by following the method described in Section 3.1.2, the AHC process (Figure 3.5) proceeds as follows: (1) The input BKs are re-assigned to the current clusters. This is done by calculating similarities between each input BK and each cluster, assigning them to the cluster which provides the highest similarity according to Equation 3.2. Segment assignment involves the computation of N times M_i similarities per AHC iteration, N being the number of input BKs, and M_i being the number of clusters at iteration i. The total number of similarity computations can be high. However, the used metric involves very fast, bit-wise operation between two binary vectors. In addition, this metric returns a very simple and meaningful real score between 0 and 1, 0 indicating total dissimilarity, and 1 indicating total similarity (i.e. the BKs being compared are identical). After the data assignment has been performed, the obtained solution at the current iteration is stored. (2) Cluster-wise BK similarities are computed using Equation 3.2 as well, the closest cluster pair is merged, and the BK for the new cluster is computed. In this step, the number of comparisons to be performed at iteration i is equal to $\frac{1}{2}M_i^2$. Then, the algorithm goes back to (1) while the current number of clusters is > 1.

3.2.2.2 Selecting the optimum number of clusters

The previous process outputs a set of N_{init} clustering solutions, each one with a decreasing number of clusters. The optimum solution has to be selected from those returned through some clustering selection algorithm. [Anguera and Bonastre, 2011] proposed an adaptation of the T-test T_s metric described in [Nguyen et al., 2008]. A given clustering solution consists of a set of clusters grouping the equal-sized segments from the input data, represented as BKs. First, the statistics of intra-cluster and inter-cluster similarity distributions (i.e. the distributions of all similarities between binary keys obtained from segments in the same cluster and between all binary keys from segments in different clusters) are calculated. Then, assuming that both distributions are Gaussian-shaped, T_s is calculated as

$$T_s = \frac{m_1 - m_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{3.3}$$

where m_1 , σ_1 , n_1 , m_2 , σ_2 and n_2 are respectively the mean, standard deviation and size of the intra-cluster and inter-cluster distributions. Finally, the clustering which maximizes the T_s value is selected.

3.3 Baseline speaker diarization system evaluation

Section 3.2 provided a detailed description of the different modules of the speaker diarization system based on binary keys. The aim of this section is to evaluate our implementation of the proposed system. This system is taken as our single-session baseline speaker diarization system. First, information of the database used in the evaluation is given. Second, the complete system set-up is given. Finally, evaluation results are provided, putting a special focus on accuracy and execution time.

As said before, the diarization system taken as a baseline in this thesis is very similar to the one described in [Delgado et al., 2014b]. However, when comparing the results reported in that work with the ones provided in this section, one will find differences in performance. These differences are mainly due to the way performance was computed. In the former work, Diarization Error Rate was computed excluding audio regions with overlapping speech. In that work, this decision was made because our system does not perform overlapping speech detection, and consequently, nothing is done in order to try to assign more than one speaker label to those regions. However, majority of current speaker diarization evaluations campaigns do not use this metric as primary metric. For this reason, in this thesis it has been decided to calculate DER including overlapping speech, so that comparisons of performance can be done with other works in the literature. Therefore, it can be expected that overlapping regions will produce a systematic increase in miss speaker time rate. A second difference with regard to [Delgado et al., 2014b] is in the Speech Activity Detection (SAD) labels used. While in the first work SAD labels were obtained automatically (by using two different methods), in this thesis we are using ground-truth SAD labels obtained from the reference speaker labels. This is done to put the focus on the different modules involved in the system, without the impact of the effects introduced by incorrect SAD labels (mainly noise introduced by false alarm errors).

3.3.1 Database

These paragraphs give an overview of the audio database used in all experiments in this thesis. Motivated by the most recent evaluation campaigns (the Spanish Albayzin campaign, and the French ESTER, ETAPE and REPERE campaigns), this work focuses on the broadcast audio domain. Early, broadcast-news-type audio was mainly used in the evaluations conducted by NIST. This kind of audio data mainly consists of planned, studio-quality speech, with low rates of overlapping speech. However, the last evaluations introduced more challenging TV/radio programs which includes spontaneous speech and a reasonable proportion of multiple speaker data. This is the case of the REPERE challenge [Kahn et al., 2012]. This evaluation campaign aims at recognizing people on multimodal conditions. One of the sub-tasks proposed is speaker diarization. For this evaluation, a multimodal (audio plus video) database was especially collected. The data was delivered in the three phases of the project (dry-run, phase 1, and phase 2). First, a dry-run corpus of 3 hours of development data and 3 hours of evaluation data was distributed to the participants. In phase 1, the dry-run development set was included in the training set, and the dry-run test set became the development test of phase 1. Finally, the phase 2 dataset consists of totally new set of data.

In this thesis, the phase 1 development and test datasets are used for all experiments. Each one consists of around 3 hours of audio excerpts extracted from different TV shows from the French BFM TV, and LCP channels. The selection was done in order to extract different speech styles. Some description of the selected TV shows is given next:

- *Top Questions* consists of extracts from parliamentary questions featuring essentially prepared speech.
- *Ca vous regarde*, *Pile et Face* and *Entre les lignes* are variants of debate programs, containing a mixture of prepared and spontaneous but relatively policed speech.
- *LCP Info* and *BFM Story* are information shows with a higher number of speakers.
- *Culture et Vous* (before called *Planete Showbiz*) is a celebrity news show with a voice over, containing majorly spontaneous speech.

Table 3.1 collects information of the development and test sets, respectively. First column contains the audio file ID. Second column ("#S") shows the number of speakers. The column "Duration" specifies the duration of the audio being evaluated (including speech and non-speech). "Speech time" column refers to the time remaining after removing the non-speech content. "Speaker time" is the sum of all speech produced by each participant speaker (it can be longer than the speech time due to overlapping speech). Finally, last column shows the percentage of speaker time which is overlapped.

It can be seen that the number of speakers per session varies from a minimum of 2-3 speakers, until a maximum of 16-17. With regard to total duration, the files have

Session name	#S	Duration	Speech time	Speaker time	%Overlap speech
REPERE Phase 1 development set					
BFMStory 2011-07-07	12	15:26.20	14:59.11	15:29.48	3.26
BFMStory 2011-07-08	6	16:16.96	16:04.63	17:02.05	5.61
BFMStory 2011-07-11	10	13:57.35	13:33.23	13:45.28	1.46
BFMStory 2011-07-13	17	15:09.93	14:37.62	14:47.40	1.10
PlaneteShowbiz 2011-06-03	7	02:11.10	02:00.11	02:20.57	14.55
PlaneteShowbiz 2011-07-05	11	03:34.17	03:16.11	03:50.68	14.98
PlaneteShowbiz 2011-07-06	9	01:46.71	01:42.36	01:57.49	12.87
PlaneteShowbiz 2011-07-08	6	01:41.08	01:36.52	01:45.20	8.24
PlaneteShowbiz 2011-07-11	4	02:14.57	02:04.84	02:07.85	2.35
PlaneteShowbiz 2011-07-12	9	02:54.91	01:43.29	01:43.52	0.21
PlaneteShowbiz 2011-07-13	11	01:43.33	01:38.34	01:48.58	9.42
CaVousRegarde 2011-05-04	5	05:29.98	05:22.45	05:57.61	9.83
CaVousRegarde 2011-05-09	7	05:03.82	04:58.05	04:59.83	0.59
CaVousRegarde 2011-07-04	4	05:05.20	05:02.79	05:04.69	0.62
EntreLesLignes 2011-07-08	5	15:00.70	14:50.22	16:27.18	9.82
LCPInfo13h30 2011-07-06	11	09:53.64	09:44.38	10:02.82	3.05
LCPInfo20h30 2011-07-05	16	10:08.79	09:47.79	09:57.65	1.64
LCPInfo20h30 2011-07-07	17	10:06.31	09:41.53	09:48.14	1.12
PileEtFace 2011-09-15	3	02:59.28	02:50.64	02:59.26	4.80
PileEtFace 2011-09-23	3	03:06.87	03:05.41	03:31.35	12.27
PileEtFace 2011-09-30	3	02:59.60	02:58.65	03:25.08	12.88
PileEtFace 2011-10-06	3	02:59.54	02:52.27	02:57.68	3.04
PileEtFace 2011-10-15	2	02:59.41	02:53.01	02:58.46	3.05
TopQuestions 2011-04-12	7	13:58.26	12:42.17	12:45.36	0.41
TopQuestions 2011-09-27	8	16:04.09	14:39.61	14:45.98	0.71
Overall	-	03:02:51	02:54:45	03:02:19	4.15
REPERE Phase 1 test set					
BFMStory 2012-01-10	6	15:04.31	14:43.67	15:44.10	6.40
BFMStory 2012-01-23	18	14:59.87	13:43.50	13:55.50	1.43
BFMStory 2012-02-14	10	18:51.50	14:23.70	14:35.92	1.39
BFMStory 2012-02-20	6	19:10.46	14:41.29	15:13.43	3.51
CultureEtVous 2012-01-13	5	01:49.39	01:42.39	01:42.60	0.20
CultureEtVous 2012-01-16	6	04:04.62	03:48.76	03:50.96	0.95
CultureEtVous 2012-01-17	16	01:43.65	01:35.04	01:35.38	0.35
CultureEtVous 2012-01-18	9	01:49.60	01:44.11	01:44.33	0.20
CultureEtVous 2012-01-19	6	01:45.27	01:26.38	01:28.81	2.73
CultureEtVous 2012-02-14	12	01:52.82	01:41.06	01:41.47	0.39
CultureEtVous 2012-02-15	14	01:41.91	01:28.11	01:29.92	2.01
CaVousRegarde 2011-12-20	7	04:59.91	04:43.98	04:45.16	0.41
CaVousRegarde 2012-01-19	5	04:57.06	04:52.29	05:10.45	5.84
CaVousRegarde 2012-01-25	5	05:04.52	05:01.55	05:44.98	12.58
EntreLesLignes 2011-12-16	5	04:57.25	04:47.95	05:23.61	11.01
EntreLesLignes 2012-01-27	5	05:03.85	04:55.59	05:14.49	6.01
EntreLesLignes 2012-05-11	5	05:01.48	04:55.20	05:01.14	1.97
LCPInfo13h30 2012-01-24	16	09:57.80	09:46.55	09:52.75	1.04
LCPInfo13h30 2012-01-25	12	09:59.59	09:34.18	09:35.35	0.20
LCPInfo13h30 2012-01-27	10	09:59.21	09:42.73	09:49.14	1.08
PileEtFace 2011-11-19	3	03:01.36	02:59.67	03:29.76	14.34
PileEtFace 2011-12-01	3	02:59.90	02:58.06	03:15.45	8.89
PileEtFace 2012-01-12	3	03:00.83	02:58.96	03:29.29	14.49
PileEtFace 2012-01-19	3	02:58.03	02:55.03	03:02.33	4.00
PileEtFace 2012-01-26	3	03:00.98	02:53.46	03:01.40	4.37
TopQuestions 2012-01-25	8	11:47.02	10:41.41	10:57.40	2.43
TopQuestions 2012-02-14	5	07:02.90	06:39.55	06:42.79	0.80
TopQuestions 2012-02-22	6	08:11.09	07:26.76	07:27.86	0.24
Overall	-	03:04:56	02:48:50	02:55:05	3.56

Table 3.1: REPERE Phase 1 development and test sets.

been selected in order to present different audio lengths, ranging from less than 2 minute files, until around 15 minute files. In particular, *Planete Showbiz/Culture et Vous*, are short excerpts between 1:41 and 4:04, containing a relatively high number of speakers given the total duration of files. This means that the speaker time of each participant speaker will be rather low, which can be challenging when estimating robust speaker models. The fourth column shows the speech time per audio file. If one compares columns 3 and 4, one can obtain the non-speech time, which can range from several seconds up to 3-4 minutes. Column 5 shows the speaker time, which is the sum of all speaker turns' durations, including regions of overlapping speech. By comparing columns 4 and 5, the proportion of overlapping speech can be obtained. This proportion is shown in column 6. The proportion of overlapping speech can be considerably high in some audio excerpts, reaching up to 14%. Other files have very low (around 1%) percentage of overlapping speech, while the rest present intermediate levels. Overlapping speech can have an important impact on diarization systems not able to detect and deal with such kind of speech in two main senses: First, if the system cannot assign additional speaker labels on such regions, overlapping speech is systematically omitted, and thus its corresponding speaker time becomes miss speech error. Second, if the system is not able to find overlapping regions, such regions will be treated as they were speech from single speakers, and therefore such data regions will be included in single-speaker clusters. This results in an insertion of noise within the clusters, which are intended to contain speech from only one speaker. The consequence of noisy clusters is a loss of accuracy of the estimated speaker models, as it has been demonstrated that noisy data affects negatively to the modeling capabilities of the majority of approaches usually used for speaker modeling.

By inspecting Table 3.1, some conclusions could be highlighted. First, the short durations and relatively high number of speakers of some audio excerpts may suppose a challenge because it has been shown that speaker modeling capabilities decreases drastically when not enough training data is available [Imseng and Friedland, 2009]. The second key point concerns with overlapping speech. Some of the files present a high proportion of overlapping speech (above 10%). Systems not dealing with overlapping speech will suffer the consequences in form of not only increments of miss error rate, but also speaker error rate due to the less accurate speaker modeling provoked by impure cluster data.

Throughout this thesis, the REPERE development and test datasets explained above are used in all experiments. The REPERE test set has been used as the primary corpus, in which all the modifications proposed have been first evaluated. After the evaluation on the REPERE test dataset, the REPERE development data set has been used as an additional dataset of unseen audio data. This database is mainly used for checking if the obtained results on the REPERE test set are consistent thought new unseen data. This election could seem contradictory, in the sense that the REPERE development set could have been used first while the REPERE test set could have been used as additional unseen data. The explanation of this choice is that, at the beginning of this research, only the test set was available for the author. Therefore, all the developments of this thesis were performed on that dataset. Later, access to the REPERE phase 1 development set was finally possible, and from then this development database has been used as an additional test set in order to validate the developments carried out in this thesis.

3.3.2 System set-up

The next paragraphs provide the experimental setup of the baseline speaker diarization system. Parameter values are essentially the ones proposed in [Delgado et al., 2014b]. All the experiments have been performed on a 2.7 GHz AMD Phenom II X6 processor with 8 GB RAM. The complete system is implemented in Matlab. The code has been optimized to the extent possible in order to take advantage of all CPU cores.

In feature extraction, standard 19-order MFCCs are computed using a 25ms window, every 10ms. For training the KBM, single Gaussian components are estimated using a 2s window in order to have sufficient data for parameter estimation. Window rate is set according to the input audio length, in order to obtain an initial pool of 2000 Gaussians. With regard to binary key estimation parameters, the top 5 Gaussian components are taken in a frame basis, and the top 20% of the components at segment level. For cluster initialization, the number of initial clusters is set to 25 in order to assure a number greater than the maximum number of speakers in the database (up to 18 in some excerpts), and the rough clustering is performed by dividing the input features in small chunks of 100ms and assigning them to the clusters through maximum likelihood. Finally, in the agglomerative clustering stage, binary keys are computed for each 1s segment, augmenting them 1s before and after, totaling 3s.

As mentioned above, performance was measured by calculating DER, which is the most standard metric for speaker diarization. In this work, the evaluation tool by LNE developed for the REPERE evaluation campaign [Galibert, 2013], and described in Section 2.4.1 was used. As explained before, the main difference with the tool developed by NIST is the way the forgiveness collar is applied. In this work, a standard forgiveness collar of 0.25s was used.

3.3.3 Evaluation

Figure 3.6 shows performance of the baseline system measured in Diarization Error Rate (DER), according to the number of components in the KBM (note that the number of KBM components defines the number of bits of the BKs), for the system output "SysOut" (i.e. the clustering returned by the clustering selection algorithm) and the optimum output "OptOut" (i.e. the best clustering in terms of DER selected manually). This last result is shown in order to set a performance ceiling (or error floor). Execution time is also provided in terms of Real Time Factor (xRT, secondary Y axis of Figure 3.6), calculated as the total time taken by the system to process the input data (excluding feature extraction) divided by the speech time (duration of the portions of the input audio labeled as speech).

Comparing DER of optimum and system outputs, it can be seen that the clustering selection algorithm is far from returning the best clustering that the system is actually able to generate (the optimum one in terms of DER). DER of system output starts decreasing when the number of components is incremented, reaching the optimum value of 18.22% at 896 components. However, from this point on, performance suffers a decrease with an increasing number of KBM components. Contrarily, the optimum output seems to converge around 11.5% DER for a number of 420 and higher.

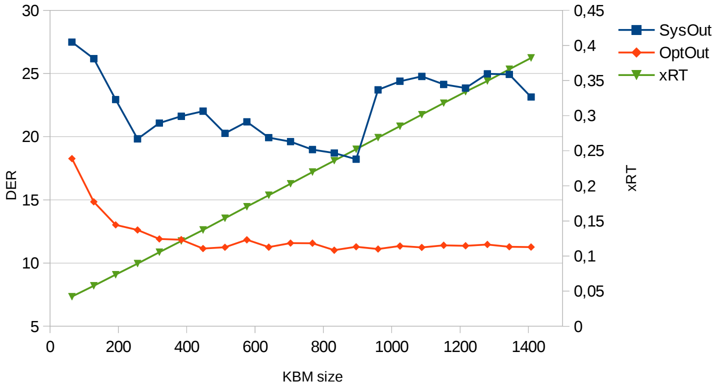


FIGURE 3.6: Baseline speaker diarization system performance, measured in DER, for the system output and the optimum output selected manually, as a function of the KBM size. Execution time measured in real-time factor (xRT) is also provided (secondary Y axis).

With regard to computation time, xRT increases linearly with the size of the KBM, ranging from 0.042 (23.8 times faster-than-real-time) with 64 components, until 0.382 (2.6 times faster-than-real-time) with 1408 components.

The best system output result is 18.22% DER with 0.252 xRT, obtained with a KBM of 896 components. With this configuration, the optimum clustering selected manually provides a DER of 11.29%. This suggests that a better clustering selection algorithm would allow to obtain clusterings closer to the optimum one.

Table 3.2 gives detailed results obtained with the best configuration found (KBM size of 896) in a per-file basis, broken down into false alarm (FA), miss error (MISS) and speaker error (SPKE). Execution times in terms of real-time factor are also provided. Finally, DER of the best clustering selected manually (DER floor) is also provided for comparison. The numbers in brackets indicate the number of reference/detected speakers.

First, let us put the focus on false alarm errors. As previously stated, ground truth SAD labels are used in this experiment. Ideally, in this situation false alarm error should be 0, as the use of perfect SAD labels derived from the reference speaker labels does not allow the system to generate speech segments where there is not speech labeled. However, this does not happen, and false alarm rates up to 1.97% are obtained. The reason of these FA rates is the way the evaluation tool applies the forgiveness collar of 250ms, as it was already shown in Section 2.4.1. Note that the collar is only applied to segments whose speaker label and its associated reference label occur at the same time. In other words, the collar does not benefit segments accounted as speaker error. This criterion results in not null false alarm, whose rates will depend on the number of segments accounted as speaker error time. For example, the audio file *CultureEtVous_7* presents a false alarm rate of 1.97%. It can be seen that the actual number of speakers in this show is 14. However, the number of speakers returned by the system is 4. This suggests that a great number of segments are accounted as speaker error, and then the forgiveness collar does not apply to those segments.

Session ID (#S)	FA	MISS	SPKE	DER (#S)	xRT	DER floor (#S)
BFMStory_1 (6)	0.76	3.59	44.85	49.20 (3)	0.15	8.28 (5)
BFMStory_2 (18)	0.90	0.53	15.46	16.89 (15)	0.15	14.07 (16)
BFMStory_3 (10)	0.26	0.72	6.71	7.69 (11)	0.15	4.75 (8)
BFMStory_4 (6)	0.06	2.33	2.90	5.28 (7)	0.15	5.28 (7)
CultureEtVous_1 (5)	0.01	0.14	26.03	26.18 (2)	0.74	14.37 (4)
CultureEtVous_2 (6)	0.85	1.55	33.20	35.60 (2)	0.34	24.03 (10)
CultureEtVous_3 (16)	1.34	1.79	32.14	35.27 (4)	0.73	30.73 (7)
CultureEtVous_4 (9)	0.03	0.85	32.49	33.36 (2)	0.73	26.93 (9)
CultureEtVous_5 (6)	1.64	3.20	14.95	19.79 (3)	0.90	19.79 (3)
CultureEtVous_6 (12)	1.46	0.85	16.20	18.50 (7)	0.74	15.10 (8)
CultureEtVous_7 (14)	1.97	2.25	40.00	44.22 (4)	0.90	38.19 (6)
CaVousRegarde_1 (7)	1.45	0.74	9.08	11.27 (4)	0.29	11.27 (4)
CaVousRegarde_2 (5)	0.43	3.28	5.03	8.74 (7)	0.28	5.90 (4)
CaVousRegarde_3 (5)	0.49	8.54	18.38	27.41 (6)	0.28	27.41 (6)
EntreLesLignes_1 (5)	0.76	3.67	32.77	37.20 (5)	0.29	31.48 (9)
EntreLesLignes_2 (5)	0.31	2.80	16.60	19.71 (4)	0.28	9.91 (5)
EntreLesLignes_3 (5)	0.61	0.79	14.24	15.64 (4)	0.28	15.64 (4)
LCPInfo13h30_1 (16)	0.50	0.72	6.72	7.94 (15)	0.18	7.94 (15)
LCPInfo13h30_2 (12)	0.17	0.02	18.42	18.62 (9)	0.19	10.72 (17)
LCPInfo13h30_3 (10)	0.35	0.93	9.77	11.05 (8)	0.18	9.98 (9)
PileEtFace_1 (3)	0.55	7.54	17.79	25.88 (3)	0.44	21.91 (2)
PileEtFace_2 (3)	0.00	7.80	7.10	14.90 (2)	0.45	11.71 (6)
PileEtFace_3 (3)	0.45	7.20	26.56	34.21 (3)	0.44	32.06 (2)
PileEtFace_4 (3)	0.40	4.03	16.33	20.77 (2)	0.44	5.95 (4)
PileEtFace_5 (3)	0.35	3.90	2.41	6.66 (3)	0.44	6.66 (3)
TopQuestions_1 (8)	0.86	0.83	9.97	11.66 (11)	0.17	2.84 (8)
TopQuestions_2 (5)	0.36	0.43	3.48	4.28 (4)	0.22	3.28 (6)
TopQuestions_3 (6)	1.52	0.00	6.99	8.51 (7)	0.21	5.03 (5)
Overall (time-weighted)	0.58	2.10	15.54	**18.22**	**0.25**	**11.29**

Table 3.2: Performance measures in a per-file basis, for a KBM size of 896, broken down into false alarm (FA), miss error (MISS), and speaker error (SPKE). xRT are also provided. Finally, DER of the best clustering outputs selected manually are shown (DER floor). (#S) indicates the number of speakers.

Second, let us focus on the miss error. Since ground truth SAD labels were used, the only source of miss error should be the speech in overlapping regions. However, it can be seen that the miss error rate does not match the percentage of overlapping speech shown in Table 3.1. Once again, this is a result of the collateral effects derived from the LNE scoring tool. Those results highlight the cost of using this evaluation method, which is optimized for overlapping speech regions, in a system which does not handle overlapping speech.

Third, let us analyze the speaker error. As we have observed poor performance of the selection of the final clustering, we put the focus on the performance ceiling (error floor). Clear evidence is the negative impact of the lack of data on speaker error. *Culture et vous* sessions are short excerpts of durations ranging from 1:41 and 4:04 (mm:ss) which contains a relatively high number of speakers. Those sessions exhibit DER floor figures of 14-38%. This may be due to the sparse data per speaker, which may be not enough data to estimate robust cluster BKs. Oppositely, processing other longer files such as *BFM story* and *Top questions* provide much better DER floors of 2-14% DER.

With regard to the number of clusters of the optimum clusterings selected manually, we observe that the number of returned clusters is not commonly the exact number of speakers. Sometimes, selecting a slightly higher or lower number of clusters results beneficial in terms of DER. This is due to the time-weighted nature of the DER metric.

	Audio file 14:43 (883s)		Audio file 01:28 (88s)	
Stage	Time (s)	xRT	Time (s)	xRT
KBM training	**66.23**	**0.0749**	**76.94**	**0.8733**
BK estimation	35.54	0.0402	3.14	0.0356
Clustering init.	0.433	0.0004	0.042	0.0004
AHC	25.29	0.0286	3.09	0.0351
Clustering selection	19.44	0.022	0.12	0.0014
Overall	146.933	**0.1661**	83.332	**0.9458**

Table 3.3: Execution time (with KBM size = 896) measured in seconds and xRT for two audio files of different durations (14:43 and 01:28, respectively), broken down into the different stages of the diarization process.

Sometimes a number of clusters slightly above the actual number of speaker may result beneficial if the extra clusters are small and the rest of cluster represent correctly their corresponding speakers.

It is interesting to compare xRT figures achieved in this work with the ones obtained in [Delgado et al., 2014b] on the RT-05 database of meeting recordings. In that work, by using a similar KBM size, xRT was around 0.18, which is considerably lower than the 0.25 obtained here. The reason is that execution time of the iterative Gaussian selection algorithm used in the training of the KBM does not depend on the duration of the input audio file, but in the initial number of components of the Gaussian pool. Regardless on the audio duration, the window rate parameter is set to get a constant number of components. Then, the desired number of Gaussian is obtained iteratively. As the real-time factor measures execution time in function of the length of the input audio, it is clear that using a fixed window length and final number of component (regardless of the audio duration) will have an increasing penalty when the audio stream is increasingly shorter.

In the case of the RT-05 database, all the excerpts have similar duration (around 12 minutes each file), whilst in the REPERE database, audio durations can range between 1 and 15 minutes. The KBM training results in a high penalty in execution time when the audio file is very short, while longer audio files are favored. Table 3.3 shows execution time figures for two audio files of different durations, broken down into the involved processes in the diarization system. It can be appreciated that the KBM training takes even more time for the short audio file than for the long one, while execution time of the rest of the processes are similar for both files. This big difference in the time required to train the KBM has a great impact on the calculation of xRT for the short audio file.

3.4 Conclusions

In this chapter, the binary key speaker modeling has been introduced for the task of speaker verification. The core element of the technique, called Binary Key Background Model (KBM), and the method for binary key extraction have been presented. Then, a speaker diarization system based on binary key speaker modeling has been described, providing details on all elements and processes involved. Finally, the implementation of the binary key based speaker diarization system used as baseline in this thesis was described and evaluated. Before providing evaluation results, the speech database used for the experiments (and also used along this thesis) is described. Finally, system set-up and experimental results were provided. The reported results in this chapter are considered to be the baseline for single-session speaker diarization in this thesis.

After analyzing the obtained experimental result, some conclusions can be extracted. The first remarkable point is the ineffectiveness of the final clustering selection algorithm. There is a considerable gap between the performance ceiling and the system outputs, and this motivates more research in this direction, as a more accurate clustering selection algorithm would result in systematic gains in performance. As explained in Section 3.2.2.2, the final clustering selection is based on the distributions of inter-cluster and intra-cluster similarities. A deeper analysis of the distributions of distances between BKs from the same speaker and between BKs from different speaker could give some insight.

A second remark concerns the use of the LNE scoring tool. Experimental results show that the methodology used by this tool is prejudicial for a system which does not handle overlapping speech, like the one described here. Interpreting miss and false alarm errors become less intuitive. Even so, it has been decided in this thesis to continue to use this scoring tool, as it enables comparison of performance with the participant systems of the REPERE challenge, where the same data and scoring tool were used.

Third, it is worth mentioning the high DER figures obtained in short audio files with a high number of speakers. A possible reason of that is the lack of enough speaker data to estimate robust BKs for those speakers. Even so, adapting values of the parameters involved in the computation of BKs in function of the audio duration could be a possible solution. Particularly, in those cases in which there is very few training data, a lower number of KBM components, or even a lower number of bits set to 1 in the BKs could better fit the needs of such situations.

A final important point to be considered is the penalty in execution time imposed by the use of constant parameters values regardless of the audio duration. In this thesis, one of the main objectives is to provide a very fast system suitable for processing large collections of data quickly. It is, therefore, crucial to speed up the process and make execution time to be dependent on the audio length.

All the mentioned above leads us to a range of research paths that need to be explored in order to find solutions to the main limitations of the system. These aspects are the ones mainly addressed in Chapter 4.

Chapter 4

Proposed improvements to single-session binary key speaker diarization

This chapter collects the research carried out in the context of this thesis in the topic of single-session speaker diarization based on the binary key speaker modeling. The term single-session speaker diarization refers to the classic task of speaker diarization on a single audio file, and it is used here in order to do a distinction between the classic problem of speaker diarization and the more recently proposed task of cross-session speaker diarization, where the recurrent speakers participating in several sessions must be assigned the same label along the complete collection being processed.

In Section 4.1, some early research is presented. First, a binary key based approach to Speech Activity Detection (SAD) is proposed and evaluated. Second, a global clustering method is adapted and applied in order to give alternatives to the faulty final clustering selection algorithm used in the baseline system. In Section 4.2, an analysis of the binary key speaker modeling is performed with the aim of acquiring knowledge about its behavior with regard to the free parameters and the amount of training data available. The resulting insight will be used to make decisions on the parameter settings and tuning. Next, Section 4.3 describes the main contributions of this work in the topic of single-session speaker diarization with binary key speaker modeling. The proposed improvements are principally addressed to speeding up speaker diarization, as well as increasing accuracy, and concern KBM training, similarity measures for binary keys and cumulative vectors, final clustering selection, and intra-session variability compensation. All the proposed improvements are evaluated and validated in Section 4.4. Finally, conclusions are given in Section 4.5.

4.1 Preliminary work

This section describes some preliminary work done in the field of binary key speaker diarization. The research covered by this section is not considered a part of the core of the contributions of this thesis. However we consider that these early research efforts are of enough interest and must not be obviated. Those research efforts include a novel approach to Speech Activity Detection based on binary keys, and the application of a global clustering scheme which partially replaces the AHC method implemented in our baseline system. As these proposals are described separately from the core contributions, they have their own experimental evaluation subsections within this section.

4.1.1 Binary key based Speech Activity Detection

As it has been pointed out in Section 2.2.2, Speech Activity Detection (SAD) is an important pre-processing tool in many speech related systems. It mainly consists in detecting the audio regions corresponding to speech and the audio regions not corresponding to speech within an input audio stream. This tool enables further systems to discard non-speech content and to focus only on speech audio.

As it has already been discussed in the state of the art, one of the most extended approaches to SAD is the called model-based approach. In this approach, models for the desired acoustic classes (normally speech and non-speech classes) are trained on appropriate labeled data. Then, in the detection phase, those models are used to classify the input signal into the different categories. A very common model based approach is that which uses GMMs for the modeling of the acoustic classes of interest, and a maximum-likelihood based decoding process (for example the Viterbi decoding) for detecting the different regions of the input signal according to the acoustic classes.

The binary key speaker modeling is based on the well-known GMM-UBM paradigm. This modeling technique has resulted to be successful in tasks such as speaker identification and verification (see Section 3.1), as well as for other tasks that are also successfully performed by the GMM-UBM paradigm (emotion recognition, speaker diarization). It is therefore reasonable to hypothesize that the task of SAD could also be successfully addressed by the binary key modeling. A second motivation to develop a binary key based SAD approach is to get a unified SAD and speaker diarization system over the same underlying modeling approach, as SAD and speaker diarization are closely related tasks that go commonly hand in hand.

Following this reasoning, we proposed a binary key based approach to SAD in [Delgado et al., 2014b]. This approach follows a model-based scheme, where a binary key is estimated for each acoustic class on training data. In the classification stage, the target input signal is converted into a sequence of binary keys, which are then assigned to the different classes according to the similarities between input and classes binary key pairs.

As in the case of binary key speaker modeling, a Binary Key Background Model (KBM) is needed in order to capture the overall acoustic space, and to act as a generator model used for estimating binary keys for both the acoustic classes and the input data.

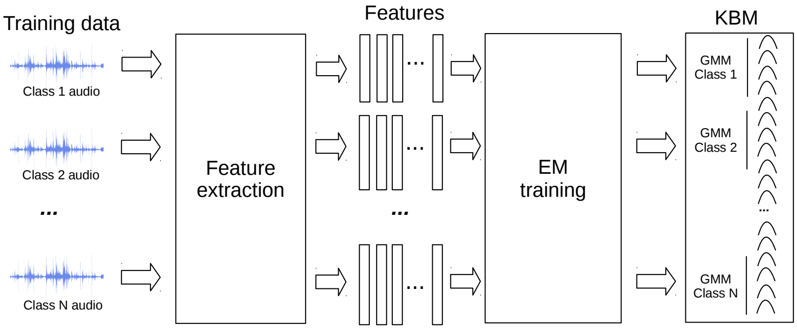


FIGURE 4.1: KBM training process for SAD.

Figure 4.1 illustrates the KBM training process. Essentially, the KBM training is done similarly to the case of speaker modeling (see Section 3.1). First, feature extraction is performed on appropriate labeled training data. Then, a GMM is trained for each acoustic class (e.g., "speech" and "non-speech") using the Expectation-Maximization algorithm. Finally, the KBM is the result of concatenating all Gaussians of the GMMs estimated for each class. Therefore, the total number of components of the KBM will be n times m, n being the number of classes, and m being the number of components of one GMM. Once the KBM is obtained, binary keys for each class can be trained on labeled data by using the binary key computation method described in Section 3.1.2.

Figure 4.2 shows the data assignment process. The input data is split into equal-sized small segments and transformed into a sequence of BKs by using the KBM and the BK estimation procedure. Finally, the input BKs are assigned to the various acoustic classes according to the class which provides the highest similarity.

Experiments and results

In order to evaluate the proposed SAD approach, experiments have been conducted on two different audio types: meeting recordings, and broadcast TV data. For meeting

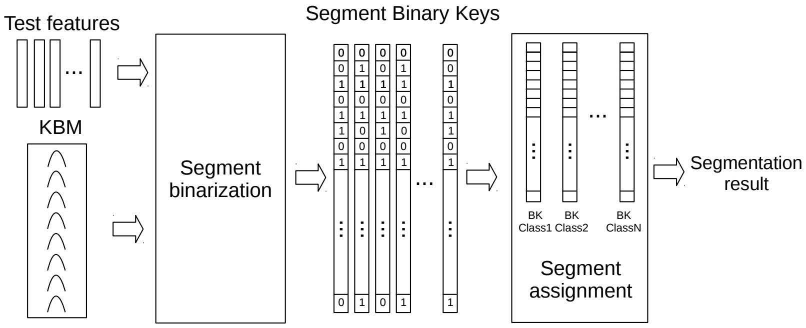


FIGURE 4.2: Input data assignment process for SAD

room audio experiments, the NIST RT05 dataset is used, whilst the REPERE phase 1 test dataset is used for TV audio experiments. The NIST RT05 database consists of a set of 10 meeting excerpts. In this experiment, the Multiple Distant Condition (MDM) of the NIST RT evaluations is used. 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 [Anguera et al., 2007]. For TV audio, the provided single channel is used without further treatment. Next, feature extraction is performed. 12-order LFCCs (Linear Frequency Cepstral Coefficients) augmented with energy and first and second derivatives (totaling a vector of 39 elements) are used.

With regard to binary key estimation parameters, the top 5 Gaussian components are taken in a per frame basis, and the top 20% of the components at segment level. As in the baseline speaker diarization system described in Chapter 3, similarities between binary key pairs are computed through the Jaccard similarity.

Concerning the acoustic classes, for meeting audio experiments only two audio classes are considered, namely speech and non-speech. However, given the richer audio variety present in broadcast audio, 4 classes are considered in the experiments with the REPERE database: speech, speech plus music, music, and narrowband speech. After data assignment, segments labeled as speech, speech plus music, and narrowband speech are renamed together as speech, while segments labeled as music are renamed as non-speech.

The data assignment is done in a similar way as in the diarization system described in Chapter 3. But given the nature of the audio been classified, the window length should be significantly smaller than the speaker window. For instance, using window rates of 1s would not capture pauses shorter than this. It can be commonly found in the literature that the minimum non-speech duration considered is 0.3s. For this reason the window rate here is set to 0.3s.

Table 4.1 shows experimental results obtained for both meeting and TV data experiments. In addition, results of a state-of-the-art HMM-based approach [Fredouille et al., 2009] are included for comparison.

It can be observed that in experiments on the NIST RT05 database, the proposed SAD system 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 in further processes 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.

A possible reason of the worse performance obtained on TV data could be the presence of more sources of acoustic variability, which could make the problem much more challenging than in the case of meeting audio, where the non-speech class usually comprises just silence and some kinds of background noises.

These preliminary experimental results validate the proposed binary key SAD system and suggest that the technique has potential for further improvement. There are a number of aspects that could be further investigated for the SAD task, and would probably lead to gains in performance. For example, alternative methods for KBM estimation could be explored in order to fit the special needs of SAD. A second potential point which

	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

Table 4.1: 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 (Classic SAD) are also included for comparison.

could be improved is the segment assignment in the segmentation stage. A Viterbi-like decoding on the binary key framework would likely help to obtain more precise segment boundaries than the fixed-length assignment method proposed here.

4.1.2 Global speaker clustering

It has been pointed out in Chapter 3 that one of the main drawbacks of the binary key speaker diarization system under study is the final clustering selection technique. That process takes all the partial clustering solutions obtained at every iteration of the AHC stage and tries to select the optimum one. The proposed technique described in Section 3.2.2.2 relies on the distributions of inter- and intra-cluster similarity populations, and aims at selecting the clustering solution whose inter- and intra-cluster similarities distributions are the most separated. However, [Anguera and Bonastre, 2011] and [Delgado et al., 2014b] reported on the weakness of this final clustering selection criterion being used, which usually does not provide the optimum clustering in terms of DER. However, [Delgado et al., 2014b] demonstrates that the diarization system is able to produce better clustering solutions than the one returned by the selection criterion. This indicates that improving the stopping criterion will systematically produce a gain in performance.

Given the weakness of the proposed technique, [Delgado et al., 2014a] explored clustering selection alternatives in order to replace the faulty method. Recently, a global optimization framework to speaker clustering was introduced in [Rouvier and Meignier, 2012]. Contrarily to classic AHC, the framework tries to find the optimum clustering in a global way, instead of relying on greedy, local-maximum-made decisions as AHC does. This technique is also able to implicitly determine the final number of clusters, thus it seems a potential candidate to replace the faulty clustering selection criterion used in our binary key diarization system.

The main argument against AHC is the greedy nature of the technique, which uses local optima to decide which cluster pair should be merged at each iteration. If an erroneous merging is produced, the error will likely be propagated through the iterations, resulting in impure clusters and, consequently, in loss of performance. Following this argument, the proposed alternative clustering method addresses the clustering as a global process, reformulated as a problem of Integer Linear Programming (ILP), in order to minimize a certain objective function, subject to a set of constraints, in a global manner.

In [Rouvier and Meignier, 2012], the authors propose to apply this global clustering method immediately after a first BIC-based AHC stage. At that point, it is assumed that the resulting clusters are pure, i.e., each cluster contains speech from a single speaker, but there may be more than one cluster referring to a same speaker. This can occur because a given speaker who is speaking over different acoustic conditions (e.g. background music, background noise) may be grouped in different clusters. It is at this point when the ILP clustering is performed in order to determine potential merges between the already generated clusters, with the aim of grouping together all those cluster referring to a same speaker. But before the ILP clustering can be applied, each cluster must be converted into a compact representation able to keep the acoustic properties of such cluster. In [Rouvier and Meignier, 2012], each cluster is converted into an i-vector which is later conditioned in order to remove the influence of the channel. Thus, given an input clustering of N clusters, a set of N i-vectors is obtained. From here on, the clusters are treated as single points, and the problem is reduced to the clustering of those new points which represent each input cluster.

Given the N points, the goal is to group them into K clusters while minimizing an objective function and meeting some constraints. Each of the points can play one of two different roles: They can act as cluster "centers", or otherwise, they must be associated to a center. At the end of the process, there will be as many clusters as centers. Intuitively, the objective function consists in minimizing the number K of clusters and the dispersion of the points within each cluster. All the points must fulfill a series of constraints: each point which is not a center can be associated with only one center and its distance to the center must be short enough (below a given threshold). Mathematically, the problem is defined as follows:

Minimize

$$\sum_{k=1}^{N} x_{k,k} + \frac{1}{D} \sum_{k=1}^{N} \sum_{j=1}^{N} d(k,j) x_{k,j} \tag{4.1}$$

Subject to

$$x_{k,j} \in \{0,1\} \qquad \forall k, \forall j \tag{4.2}$$

$$\sum_{k=1}^{N} x_{k,j} = 1 \qquad \forall j \tag{4.3}$$

$$x_{k,j} - x_{k,k} \leq 0 \qquad \forall k, \forall j \tag{4.4}$$

$$d(k,j) x_{k,j} < \delta \qquad \forall k, \forall j \tag{4.5}$$

Equation 4.1 is the objective function to be minimized. The first term $\sum_{k=1}^{N} x_{k,k}$ defines the number of clusters, while the second term $\frac{1}{D} \sum_{k=1}^{N} \sum_{j=1}^{N} d(k,j) x_{k,j}$ defines the dispersion of the points within each cluster. The binary variable $x_{k,k}$ is equal to 1 if the point k is a center. $d(k,j)$ is the distance between points k and j. D is a normalization factor equal to the longest distance $d(k, j)$ for all k and j. The binary variable $x_{k,j}$ is set to 1 if point j is associated with center k. Equation 4.2 forces variable $x_{k,j}$ to be binary. Equation 4.3 ensures that each point j is associated with only one center k. Equation 4.4 ensures that if a point j is assigned to center k, then k is a center. Finally, each point j associated with center k must have a distance shorter than a threshold δ (Equation 4.5).

It can be seen that this clustering framework could be applicable to any other modeling approach able to represent a speaker cluster as a single vector or point. Therefore, the adaptation of the ILP clustering to our binary key speaker diarization system is straightforward. Instead of using i-vectors to represent clusters, binary keys can be used instead. With regard to the distance similarity metric, the distance D between two BKs can be defined as $D(k, j) = 1 - S(k, j)$, where $S(k, j)$ is the similarity measure between BKs defined in Section 3.1.3.

The proposed ILP clustering requires an input clustering to start the process. This input clustering will be the result of applying a given number of iterations of the AHC binary key diarization system. Each cluster will be represented as a BK, extracted following the method for BK computation described in Section 3.1.2. Ideally, the input clusters should be as pure as possible, since the ILP clustering method is not able to re-allocate misclassified data, so the errors will be propagated to the resulting clusters. For this reason, an analysis of cluster purity across the AHC iterations will be conducted previous to the application of the global clustering in order to obtain a good set of pure speaker clusters.

4.1.2.1 Experiments and results

In this section, experimental setup and results for two different experiments are described. Firstly, a study of cluster purity across the iterations of the baseline AHC is performed. Secondly, the resulting purest clusterings from the first analysis are taken to be used as input clusters to test the ILP global clustering.

In order to focus on speaker clustering and stopping criterion, perfect SAD labels are used here in order to work in a controlled environment and to minimize the impact of SAD errors, especially false alarm errors which result in the inclusion of noise within the input speech signal.

As throughout all this thesis, the REPERE phase 1 test dataset of TV data is used on the experiments. As for the experimental setup, the settings of the parameters involved in the binary key speaker diarization system are the same used in the experiments of the baseline system described in Section 3.3.2. However, in this case, the size of the KBM was set to 896 components.

Searching the purest clusters

It has been mentioned that the ILP clustering approach requires an initial set of speaker clusters, and that those clusters should be highly pure since the further clustering stage is not able to reallocate misclassified data within the initial clusters. For this reason, first

Session ID	#spk	Highest purity		SysOut purity	
		#C	Purity	#C	Purity
BFMTV_BFMStory_1	6	19	0.910	8	0.883
BFMTV_BFMStory_2	18	18	0.891	18	0.891
BFMTV_BFMStory_3	10	19	0.951	13	0.937
BFMTV_BFMStory_4	6	11	0.962	6	0.952
BFMTV_CultureEtVous_1	5	14	0.950	4	0.891
BFMTV_CultureEtVous_2	6	10	0.925	4	0.776
BFMTV_CultureEtVous_3	16	22	0.904	5	0.744
BFMTV_CultureEtVous_4	9	22	0.890	2	0.640
BFMTV_CultureEtVous_5	6	21	0.905	3	0.823
BFMTV_CultureEtVous_6	12	18	0.870	5	0.700
BFMTV_CultureEtVous_7	14	18	0.839	6	0.701
LCP_CaVousRegarde_1	7	23	0.925	4	0.823
LCP_CaVousRegarde_2	5	7	0.938	4	0.903
LCP_CaVousRegarde_3	5	21	0.950	6	0.836
LCP_EntreLesLignes_1	5	15	0.919	10	0.909
LCP_EntreLesLignes_2	5	27	0.932	6	0.891
LCP_EntreLesLignes_3	5	15	0.945	3	0.823
LCP_LCPInfo13h30_1	16	21	0.890	14	0.882
LCP_LCPInfo13h30_2	12	18	0.951	11	0.890
LCP_LCPInfo13h30_3	10	13	0.905	7	0.871
LCP_PileEtFace_1	3	14	0.921	3	0.821
LCP_PileEtFace_2	3	15	0.954	3	0.864
LCP_PileEtFace_3	3	9	0.910	3	0.797
LCP_PileEtFace_4	3	7	1.000	6	0.988
LCP_PileEtFace_5	3	18	0.936	3	0.912
LCP_TopQuestions_1	8	22	0.987	8	0.976
LCP_TopQuestions_2	5	15	0.985	3	0.914
LCP_TopQuestions_3	6	11	0.973	5	0.948

Table 4.2: Cluster purity analysis broken down into sessions, for the purest clustering solution ("Highest purity"), and for the optimum clustering in terms of DER ("SysOut purity"). The number of clusters (#C) and the actual number of speakers #spk are also provided.

the average cluster purity is calculated among the iterations in order to find the partial clustering solution with highest purity.

In a given cluster, there should always be a majority speaker, who is the one with most speaker time within the cluster. Considering this speaker as the "main" speaker, cluster purity can be calculated as the ratio between the cluster time assigned to the main speaker and the total cluster time. The total purity of a given clustering solution could be calculated as the average purity of all clusters of that clustering solution. However, purity of clusters of different sizes does not affect, globally speaking, in the same way to the actual purity of a solution. Due to this fact, the calculation of a time-weighted purity measure is proposed instead by taking into account cluster sizes. The final time-weighted cluster purity is calculated as the cluster purity multiplied by the cluster length, and divided by the total duration of the test speech time. Finally, the time-weighted purity for a whole clustering can be obtained as the average of the time-weighted purity of all clusters in the clustering.

Generally, purity should start to increase after a few iterations of AHC and will start to decrease when incorrect merges are produced. In Table 4.2, clustering purity is shown for two different clusterings: the one providing highest purity ("highest purity" columns) and the one producing lowest DER ("SysOut purity" columns). In addition, the number of clusters (#C) is provided. Generally, purities reach the highest values in early iterations of the AHC (with a number of clusters significantly higher than the

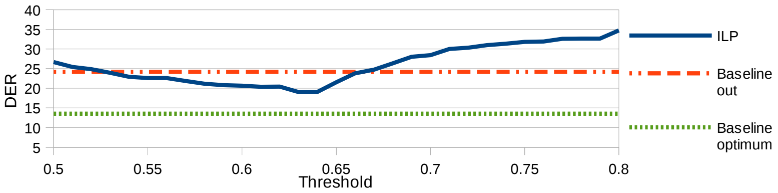


FIGURE 4.3: Overall DER trend of the ILP clustering while varying the threshold δ. Overall DER of the baseline system output (Baseline out) and optimum clusterings (Baseline optimum) are also provided for comparison.

optimum clustering), although the exact iteration is showed to be quite dependent on the session. With regard to the system output, as it could be expected, purity suffers some decrease compared to the highest purity values, probably due to incorrect data assignments and cluster merges.

ILP clustering results

Once the clustering solutions with highest purity have been found for each session, those are taken as starting point for the global ILP clustering. Figure 4.3 shows the resulting DER when varying the value of the threshold δ. For reference, DER returned by the faulty clustering selection algorithm of the baseline AHC system (orange) and DER of the optimum clustering of the AHC system selected manually (green) are provided. It can be appreciated how DER starts decreasing when increasing the threshold, reaching a minimum DER value of 19.05% with $\delta = 0.63$. From this point, DER starts increasing as δ becomes higher. This result can be interpreted as follows: when δ is low, the number of cluster merges is reduced, as there are fewer clusters which fulfils the minimum distance which allows them to be associated to a center. In other words, the lower the threshold is, the higher the number of initial clusters which form a cluster by their own is. When δ is bigger, the algorithm allows the merge of clusters that are more dissimilar, thus DER increases as more incorrect merges between clusters are allowed. It can be seen how the ILP clustering with $\delta = 0.63$ outperforms the baseline system result provided by the clustering selection algorithm ($DER = 19.06\%$ of ILP against $DER = 24.17$ of the baseline). However, this result is still far from the performance ceiling set by the optimum clustering solutions selected by hand ($DER = 13.55$).

In order to check the robustness of the threshold δ, optimum results in a per-session basis are provided in Table 4.3. In this table, DER is provided by setting the threshold to the optimum value for each session. The optimum values of the threshold are collected by the column "δ_{opt}". Additionally, the number of clusters of the resulting solutions is shown in column #C. Finally, optimum DER figures of the baseline AHC optimum ("Baseline DER") solutions selected manually are provided for comparison. It can be clearly appreciated how the optimum threshold value is totally dependent on the session being processed, and that for those manually-tuned thresholds, DER of the ILP clustering provides very similar performance as the optimum outputs selected manually. This results suggest that using a common threshold for all the sessions penalizes performance. A possible explanation to the dependence of the threshold on the session is that the distance

Session ID	#spk	δ_{opt}	#C	DER (%)	Baseline DER (%)
BFMTV BFMStory 1	6	0.63	5	9.62	9.71
BFMTV BFMStory 2	18	0.5	18	20.11	20.35
BFMTV BFMStory 3	10	0.57	11	8.08	9.25
BFMTV BFMStory 4	6	0.62	7	4.43	4.31
BFMTV CultureEtVous	5	0.77	2	21.12	12.98
BFMTV CultureEtVous	6	0.81	3	15.82	21.76
BFMTV CultureEtVous	16	0.77	4	37.74	30.53
BFMTV CultureEtVous	9	0.83	4	41.01	35.46
BFMTV CultureEtVous	6	0.77	4	40.51	15.42
BFMTV CultureEtVous	12	0.76	4	29.87	28.82
BFMTV CultureEtVous	14	0.77	4	35.79	37
LCP CaVousRegarde 1	7	0.69	4	11.7	17.7
LCP CaVousRegarde 2	5	0.63	4	7.99	7.99
LCP CaVousRegarde 3	5	0.69	9	36.52	36.49
LCP EntreLesLignes 1	5	0.63	11	26.14	28.61
LCP EntreLesLignes 2	5	0.68	5	8.34	10.06
LCP EntreLesLignes 3	5	0.77	3	17.33	16.66
LCP LCPInfo13h30 1	16	0.52	15	12.14	12.06
LCP LCPInfo13h30 2	12	0.53	11	15.92	11.57
LCP LCPInfo13h30 3	10	0.57	6	15.25	13.61
LCP PileEtFace 1	3	0.77	2	22.68	17.45
LCP PileEtFace 2	3	0.74	3	16.82	17.34
LCP PileEtFace 3	3	0.7	5	20.15	22.52
LCP PileEtFace 4	3	0.63	6	8.19	9.09
LCP PileEtFace 5	3	0.74	2	11.85	8.11
LCP TopQuestions 1	8	0.63	8	4.16	4.34
LCP TopQuestions 2	5	0.72	4	4.18	9.41
LCP TopQuestions 3	6	0.66	5	3.29	5.85
Overal	-	-	-	13.55	13.55

Table 4.3: Results of ILP clustering experiments using the purest clusterings from Table 4.2 as input clusters. For each session, values of the optimum threshold δ_{opt}, the resulting number of clusters #C, and DER are shown. The actual number of speakers per session #spk and the baseline DER are also provided.

ranges for each session can be very varying. Note that for each session, a particular KBM is trained on it in order to estimate BKs, so it is reasonable to think that the ranges of distances between binary keys are not consistent from a collection-wide point of view. Consequently, the use of a common threshold for all sessions in the database could not be a good choice.

To sum up, it could be stated that the ILP clustering method is suitable in our binary key speaker diarization system, but only partially. First, an analysis of cluster purity is required in order to select the best clustering candidates as input for the global clustering selection. This reduces the practical use of the approach in real-life situations. Secondly, it does not seem a good idea to use a common threshold along different audio files since the ranges of distances between clusters can vary significantly from session to session. In this regard, although the original clustering selection algorithm does not work correctly, that kind of approach based on ratios between intra- and inter-cluster similarities could be preferred, as they do not rely on hard thresholds.

4.2 Analysis of binary key speaker modeling

In this section, an analysis of the binary key speaker modeling is presented. This analysis is intended to study the behavior of the technique under different conditions principally related to the amount of training data available to estimate BKs.

The first part of the analysis focus on the study of the KBM, and particularly, on the Gaussian component selection algorithm involved. The second part refers to the binary keys extracted using the KBM and speech from different speakers.

This analysis is done in a controlled environment. Several experiments are performed on a database of speech from seven synthetic voices, comprising 5 male speakers and 2 female speakers. Those speech files were generated using the TTS engine by Telefonica Research. All the speech files were generated using the same source text for all the synthetic voices, consisting on an excerpt of the first chapter of the Spanish Constitution. The amount of data per speaker is around 20 minutes of speech. In this way, we set a controlled framework in which we can perform an analysis of the binary key speaker modeling technique.

4.2.1 KBM analysis

The Binary Key Background Model (KBM) constitutes the core of the binary key speaker modeling. As it has been explained in Section 3.2.1.1, the KBM is some kind of a UBM model. First, a pool of single Gaussian components is extracted from the input signal. Then, a subset of those Gaussians are selected iteratively in order to get the most discriminative and complementary ones. The process relies on the KL2 distance between Gaussian components. This iterative process is intended to return a set of Gaussians which represents the overall speaker space. Therefore, it is expected that the resulting KBM contains Gaussians trained on all the participant speakers in the input audio stream, in order to provide a good representation of all of them, and regardless of the amount of audio available for each one.

KL2 has been widely used for measuring how similar/different two Gaussians are. In order to check its effectiveness, all the available speech data of our TTS database has been concatenated to obtain a single audio stream of around 140 minutes. Then, single Gaussians are trained using a sliding window of 2s, with a window rate of 0.5s. The result is a pool of 17469 Gaussians. Finally, pair-wise KL2 distances are computed for all possible Gaussian pairs. The obtained distances are stored as within-speaker distances if the Gaussian pair belongs to the same speaker, and as between-speaker distances if each Gaussian in the pair belongs to a different speaker. Figure 4.4 shows the histograms of within-speaker and between-speaker Gaussians distances. As expected, the within-speaker and between speaker distances distributions are fairly separated, and therefore KL2 distance is presented as a suitable metric for comparing Gaussian components to be used in the component selection stage of the KBM training.

The next experiment aims to analyze the percentage of components selected for each speaker under different conditions of speaker data availability. Figure 4.5 shows the percentage of Gaussians belonging to each speaker selected by the algorithm, for different

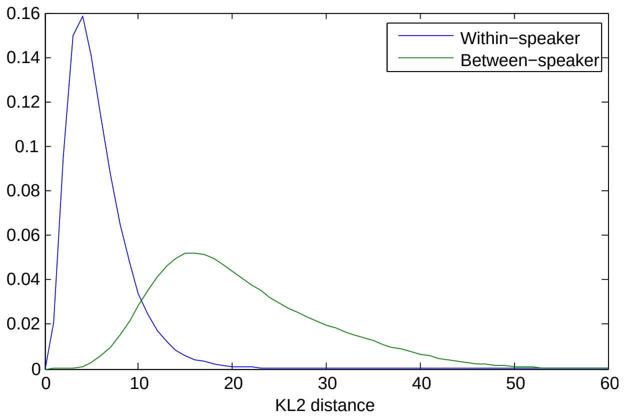
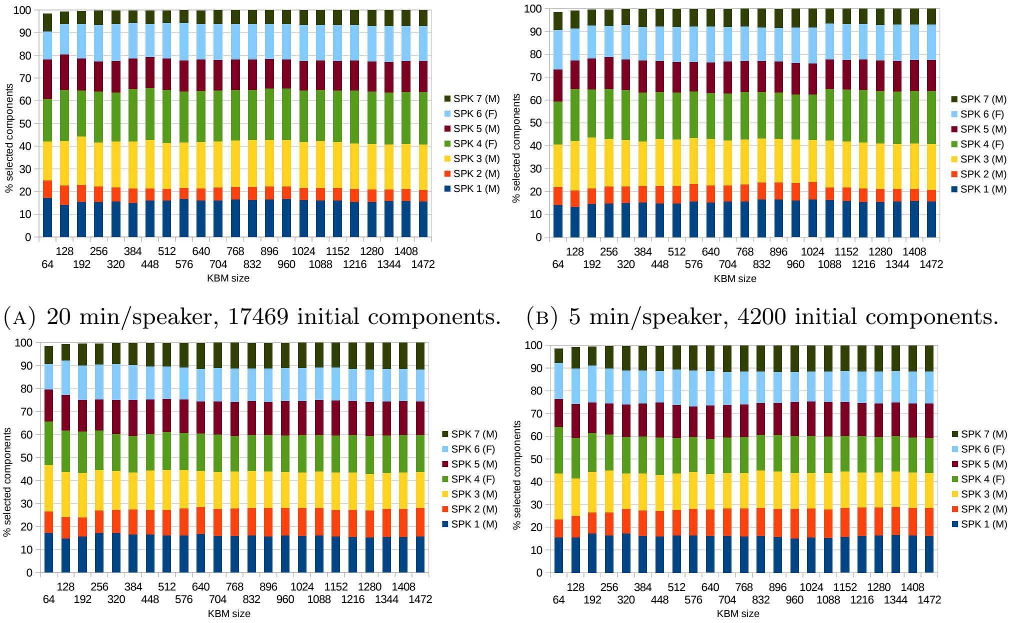


FIGURE 4.4: Normalized histograms of KL2 distances between within-speaker Gaussians (blue), and between-speaker Gaussians (green).



(A) 20 min/speaker, 17469 initial components.

(B) 5 min/speaker, 4200 initial components.

(C) 1 min/speaker, 2100 initial components.

(D) 30 s/speaker, 2100 initial components.

FIGURE 4.5: Percentage of components selected per speaker in the KBM, with regard to the KBM size, using different amounts of training data (20m, 5m, 1m, and 30s per speaker). (F) and (M) indicate male and female speakers, respectively

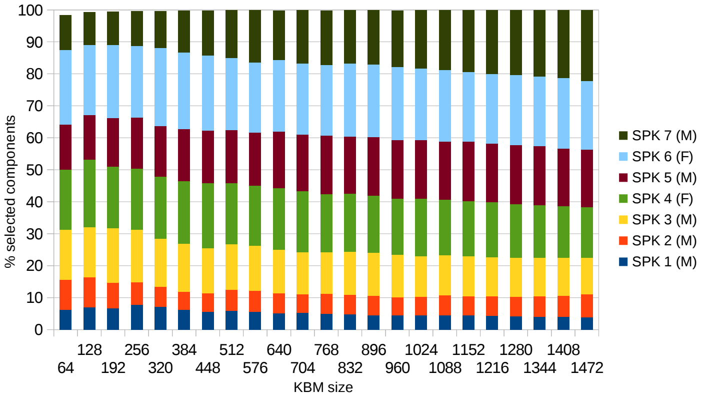


FIGURE 4.6: Percentage of components selected per speaker in the KBM, with regard to the KBM size, using a different amount of training data for each speaker: 30s, 60s, 90s, 120s, 150s, 180s, and 210s for speakers 1, 2, 3, 4, 5, 6, and 7, respectively.

amounts of training data per speaker: 20 minutes/speaker, 5 minutes/speaker, 1 minute/speaker, and 30 s/speaker. The result is shown for different KBM sizes (X axis). Note that the initial size of the Gaussian pool varies depending on the amount of training data and the window rate, which has been adapted to each case in order to produce a number of Gaussian high enough. We can observe that in the case of 20 minutes/speaker (Figure 4.5a), the proportion of selected components per speaker is in some way biased, favoring some speakers (SPK3, SKP4) at the expense of others (SPK2, SPK7) (interestingly, one of the favored speakers is one of the two female speakers present). However, this proportion keeps quite stable across different KBM sizes, and all the speakers are reasonably well represented in the resulting KBMs. When we reduce the amount of training data, it seems that the proportion of selected components per speaker gets more uniform. It could be argued that the proportion of selected components per speaker converges to the actual number of Gaussians per speaker in the input pool. However, it can be seen, for instance in Figure 4.5c, that the proportions are reasonably uniform even for small KBM sizes. Then, we could state that the component selection algorithm plays the assigned role successfully, at least in this controlled environment.

In real cases, it is quite unlikely that the amount of data per speaker is exactly the same for all speakers. In this regard, Figure 4.6 shows the proportion of selected Gaussians per speaker when the amount of training data is different for each speaker. In this experiment, the amount of training data is incremented in chunks of 30s for each speaker, starting with SPK1 with 30s, SPK2 with 60s, until SPK7 with 210s. It can be appreciated how the selection algorithm is quite insensitive to the actual proportions of data per speaker for small KBM sizes, providing reasonably balanced proportions of components per speaker.

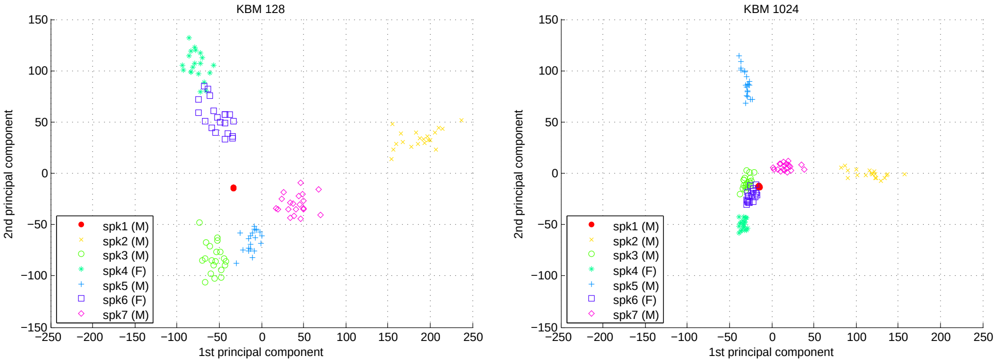


Figure 4.7: BKs from several speakers extracted using KBMs of 128 (left) and 1024 (right) Gaussians, projected on the two first directions obtained by PCA.

4.2.2 Binary key analysis

Once the speaker coverage of the KBM has been assessed, it is necessary to evaluate if binary keys generated from the KBM are discriminant enough to distinguish between speakers.

First, it would be desirable to examine the BKs graphically. However, it is well known that multidimensional data cannot be easily represented graphically, unless some sort of dimension reduction is applied. Principal Component Analysis (PCA) is a method classically used to perform a dimension reduction of possibly correlated variables into a new space defined by the called “principal components”. This transformation is done in such a way that the first principal components describe the maximum possible variance. In this experiment, KBMs of size 128 and 1024 are trained using all the available data in the TTS database (20 minutes per speaker). Then, BKs are extracted from 20 segments of 3s per speaker, totaling 140 BKs. In the BK computation, the number of top components per frame is set to 5, and the percentage of bits set to 1 is 20%. Finally, PCA is performed on the BKs set. Figure 4.7 shows the BKs extracted from KBMs of 128 and 1024 Gaussian components, projected into the two first principal components. In both cases it can be seen how the BKs from a given speaker are close together, providing fairly good separability between speakers, even using only the two first PCA directions. On one hand, it is observed that BKs from the same speaker are closer together in the case of a KBM of 1024 components. On the other hand, BKs from different speakers are also closer.

Assuming that the BKs are discriminative enough, a meaningful similarity measure between BKs is required. In our speaker diarization system based on binary keys, the Jaccard similarity was used and it has been proven empirically to be effective in the task. The next experiments focus on assessing the discriminability of the Jaccard similarity computed over BKs estimated on different amounts of speaker data. Figure 4.8 shows the normalized histograms of between-speaker and within-speaker similarities between BKs estimated on segments of 10s duration for different KBM sizes. We observe that the two populations are well separated. This indicates that threshold-based decisions on the

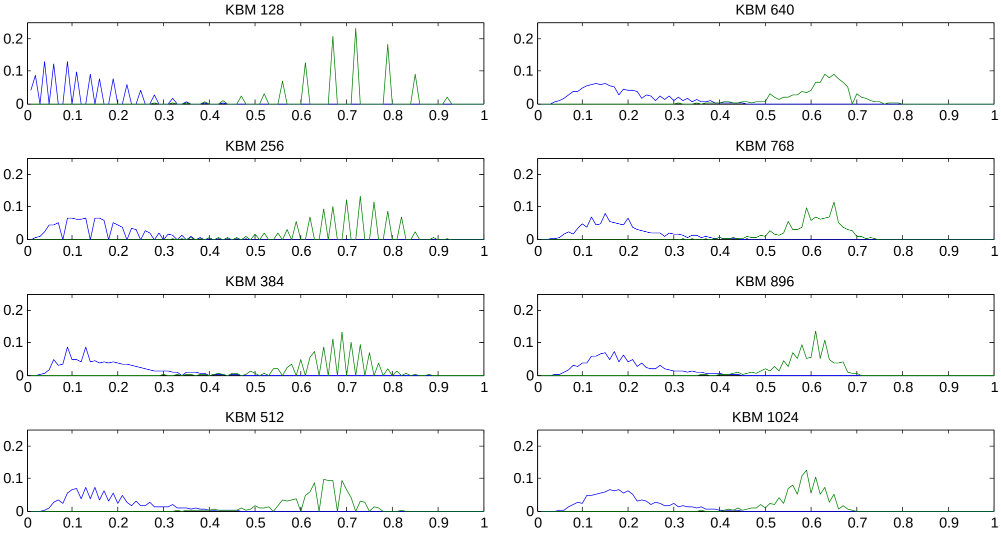


FIGURE 4.8: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **Jaccard similarity** between BKs calculated on segments of **10s duration**, for different KBM sizes.

similarities between BKs pairs could be made. However, there are some differences for the different sizes of the KBM. For small KBMs, the similarities are reduced to a small set of discrete values. This is due to the formulation of the Jaccard similarity, which calculates the percentage of nonzero coordinates that differ. When the KBM is small the possible combinations of bits are reduced and this lead to the set of discrete similarity values. When using bigger KBMs, the range of similarities becomes more continuous. In addition the width of the between- and within-speaker similarities is larger for small BKs than for long BKs. Intuitively, these results confirm the reading of Figure 4.7: within-speaker BKs derived from small KBMs seem to be more disperse than BKs derived from bigger KBMs. Finally, note that the maximum possible value of the Jaccard similarity is 1. However, the mean similarity between BKs of a same speaker is around 0.6-0.7, depending on the size of the KBM.

It seems that binary keys computed on segments of 10s are quite discriminative. However it is interesting to check whether the exhibited discriminability suffers when the BKs are estimated from shorter speech segments. Figure 4.9 replicated the results of Figure 4.8, but using BKs trained on 1s segments. Indeed, histograms of between-speaker and within-speaker similarities are now much more overlapped, and separability becomes less obvious. In this case, separating speakers through thresholding may not be adequate.

In order to find the minimum amount of data required to estimate good BKs, Figure 4.10 replicates the experiment by using 3s segments. In this case, separability is again reasonably good. The results confirm the original decision of using 3s segments for the conversion of the input data into a sequence of binary keys in a speaker diarization task. However, the within-speaker similarities mean is even lower than in the case of 10s

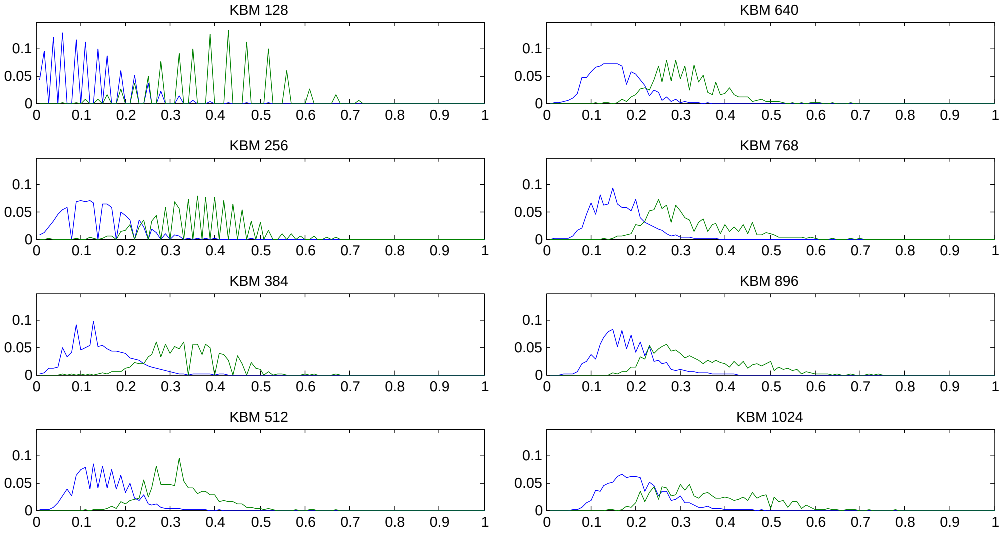


FIGURE 4.9: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **Jaccard similarity** between BKs calculated on segments of **1s duration**, for different KBM sizes.

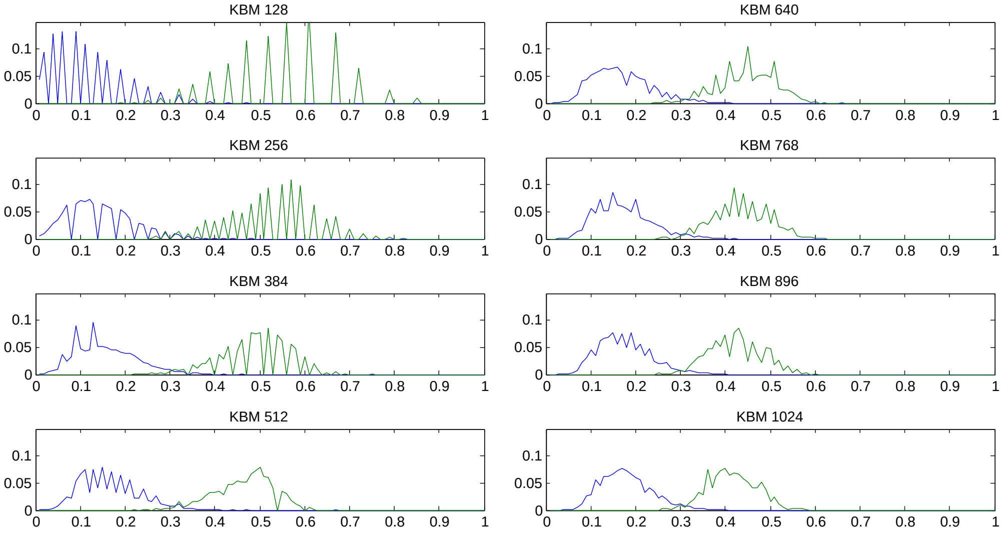
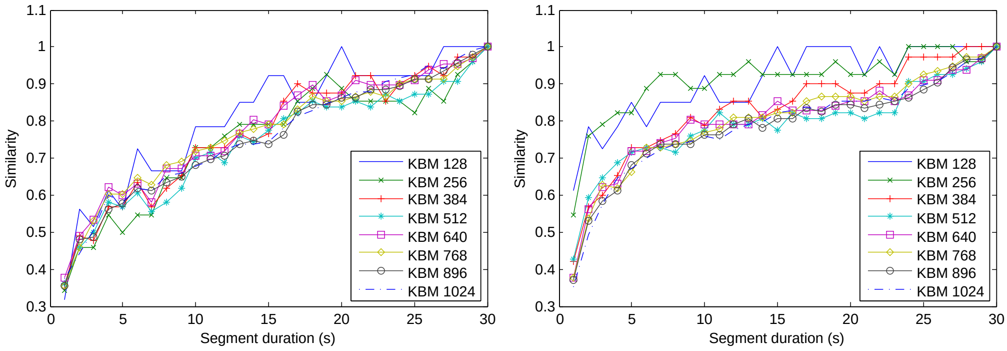


FIGURE 4.10: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **Jaccard similarity** between BKs calculated on segments of **3s duration**, for different KBM sizes.



(A) Female speaker.

(B) Male speaker.

FIGURE 4.11: Evolution of the Jaccard similarity between a BK from an increasingly longer segment (ranging from 1s to 30s) and a BK extracted from a 30s segment, using KBMs of different sizes, for a female (4.11a) and a male (4.11b) speakers.

segments. Although the two similarity populations are quite well separated. It seems reasonable to think that a similarity measure which favors pairs of within-speaker BKs could contribute to better discriminate between speakers.

The last experiment focus on evaluating the evolution of the Jaccard distance between BKs when they have been estimated on segments of different durations. This is done in order to replicate a common situation in the speaker diarization system based on binary keys, where some binary keys are derived from complete speaker clusters, and others are derived from small segments (input data BKs). In our system, cluster BKs are computed using all the data contained on the cluster. Therefore, the amount of data used in the estimation can vary from several seconds to several minutes. Figure 4.11 illustrates this situation for two different speakers (female speaker, left figure, and male speaker, right figure): A BK estimated on a variable-length segment (X axis) is compared with a BK trained on 30 seconds of speaker data. Y axis shows the Jaccard similarity between the variable-length BK and the 30s BK (which plays the role of a speaker cluster BK). These figures are calculated for different KBM sizes. We observe that similarities between BKs of very different lengths exhibit rather low values, and that the similarities increase with the amount of data used. Only for similar segment lengths (15s and above), similarities are reasonably high. However, this might not be necessarily negative since the mean of the between-speaker similarities is around 0.4-0.5 for BKs derived from 3s segments, as shown in Figure 4.10. This suggests that similarities between BKs of the same speaker to its cluster BK will still be higher than similarities between BKs to a cluster BK from a different speaker. Figure 4.12 shows the histograms of within-speaker and between-speaker similarities measured between BKs extracted from segments of 3s and speaker cluster BKs trained on 30s of speaker data, for a KBM size of 512. It can be seen how the two similarity populations are well separated.

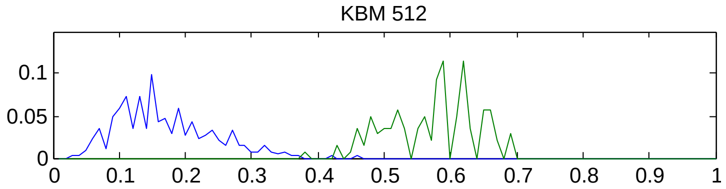


FIGURE 4.12: Normalized histogram of within-speaker and between-speaker similarities measured between BKs extracted from segments of 3s and speaker cluster BKs trained on 30s of speaker data, for a KBM size of 512.

4.2.3 Summary of the analysis

After this analysis of the binary key speaker modeling, some conclusions can be extracted.

The KBM presents reasonably good coverage of the speakers present in the input audio stream. This indicates that the iterative component selection algorithm based on the KL2 distance performs its task adequately.

The effectiveness of the KBM training process is also confirmed by the extracted BKs, which present very good discriminability between speakers. The graphic representation of binary keys through PCA suggests a good separation between speakers, even using only the two first principal components. The calculated histograms of within-speaker and between-speaker BKs similarities support this hypothesis, as it is shown that their populations are well separated, and a threshold could be effectively used to make decisions. However, a robust BK requires a minimum of training data to be estimated. It has been shown that 3 seconds of speech is enough to estimate discriminative BKs. It has also been demonstrated that BKs estimated on segments of different durations can be effectively compared. This fact enables the possibility of comparing cluster BKs with small segment BKs, which is very useful in a speaker diarization scenario. Finally, the Jaccard similarity calculated on within-speaker BKs presents rather low values, far from the maximum possible value of 1. Possibly, a similarity metric which takes advantage of the whole range of (0,1) could provide additional speaker discriminability.

4.3 Proposed improvements

This section describes the main contributions of this thesis to the binary key speaker diarization approach described in Chapter 3. All the introduced modifications have been designed in order to accomplish two main goals: first, to further speed up the baseline system, and second, to improve diarization performance.

In Section 3.3.3, the iterative Gaussian component selection algorithm used in the KBM training was identified as the main bottleneck (in terms of execution time) of the entire speaker diarization system. In Section 4.3.1 modifications to the original algorithm are proposed in order to lighten the process while preserving the effectiveness of the Gaussian selection process.

With regard to system performance, the improvement are proposed in different blocks of the system: similarity measures between cumulative vectors (Section 4.3.2), final clustering selection (Section 4.3.3), and session variability compensation (Section 4.3.4).

After all the proposed improvements are described in their corresponding sections, all them are evaluated and validated experimentally in Section 4.4.

4.3.1 Speeding up the KBM training

We showed in Section 3.3.3 that the main bottleneck of our approach is the KBM training process. It has been shown that this part penalizes execution time, especially when the input audio file is short. Speeding up this stage should result in appreciable gains in overall execution time.

As it was explained in Section 3.2.1.1, the Gaussian component selection is done in an iterative way by calculating the KL2 distance between the already selected Gaussians and the remaining ones. KL2 provides a measure of how different two probability distributions are. Let us recall the definition D_{KL2}, namely “Symmetric Kullback-Leibler Divergence”, of Gaussian distributions P and Q, which is defined by Equation 4.6 as

$$D_{KL2}(P||Q) = D_{KL}(P||Q) + D_{KL}(Q||P) \tag{4.6}$$

where $D_{KL}(P||Q)$ is the Kullback-Leibler divergence of distributions P and Q. KL is a non-symmetric measure, and this is the reason why KL2 is used instead. D_{KL} for multivariate normal distributions is defined by Equation 4.7 as

$$KL(P||Q) = \frac{1}{2}\left(\mathrm{tr}(\Sigma_Q^{-1}\Sigma_P) + (\mu_Q - \mu_P)^t\Sigma_P^{-1}(\mu_Q - \mu_P) - k - \ln\left(\frac{\det\Sigma_P}{\det\Sigma_Q}\right)\right) \tag{4.7}$$

where Σ_P, μ_P, Σ_Q, and μ_Q are the covariance matrices and mean vectors of distributions P and Q, respectively, and k is the dimension of the data.

As shown in Equation 4.7, computation of KL involves a series of matrix operations, including traces, inversions and determinants. KL2 has been largely used as cluster similarity measure and for speaker segmentation in speaker diarization, as a metric of how different two distributions are. However, it could be argued if a simpler and faster, yet useful, method could be used instead within the KBM training. As the aim of the iterative Gaussian selection process is to select the most discriminant and complementary ones, maybe calculating distances between the means of the Gaussians (centroids of the distributions) could be powerful enough to select the most dissimilar components, without taking into account the covariance matrices of the distributions. Following this reasoning, we propose the use of the cosine distance between the Gaussian mean vectors as distance metric between Gaussian components. The cosine distance $D_{cos}(a, b)$ is defined as $D_{cos}(a, b) = 1 - S_{cos}(a, b)$, where $S_{cos}(a, b)$ is the cosine similarity between two vectors a and b, defined by Equation 4.8 as

$$S_{cos}(a, b) = \frac{a \cdot b}{||a|| ||b||} \tag{4.8}$$

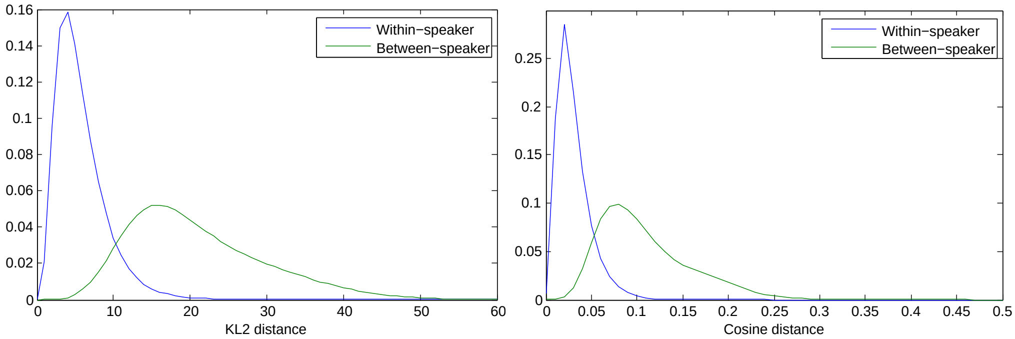


Figure 4.13: Normalized histograms of KL2 and cosine distances between within-speaker Gaussians (blue), and between-speaker Gaussians (green).

The cosine similarity formulation is considerably simpler than the KL2 one, and therefore its computation is faster. In order to check the feasibility of the Gaussian component selection through the cosine distance, an analysis similar to the one performed in Section 4.2.1 could be conducted. That analysis aimed to study the proportion of Gaussian components selected for each speaker present in the training data.

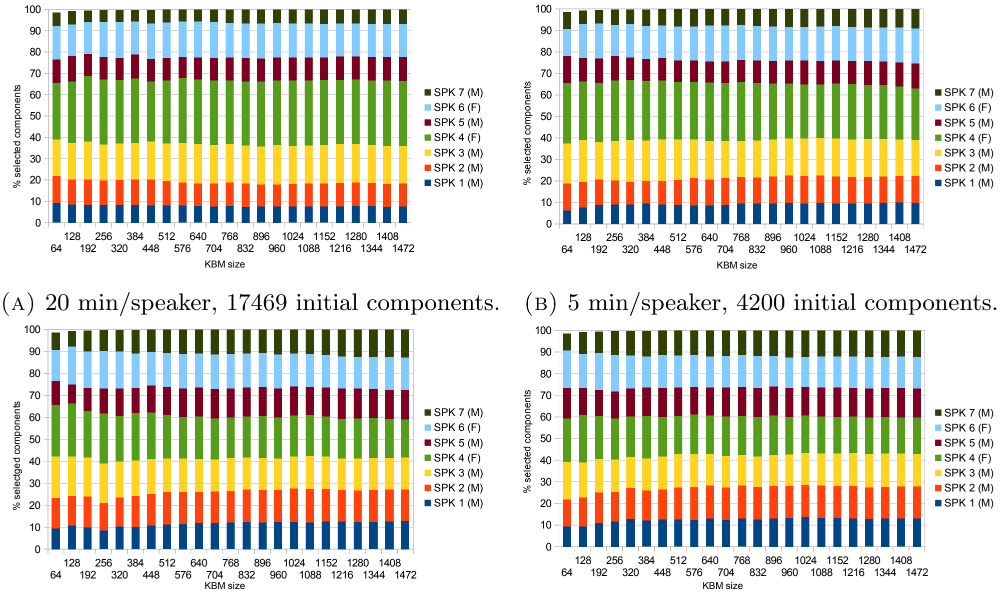
The first aspect that must be assessed is whether the cosine distance between Gaussian mean vectors is effective for measuring Gaussian closeness. Figure 4.13 shows the normalized histograms of within-speaker and between-speaker KL2 and cosine distances between Gaussian mean vectors. It can be observed that the shape of both histograms are quite similar. This indicates that, in principle, the cosine distance could be effective for measuring Gaussian similarities as KL2 is.

Indeed, the proportions of selected components per speaker by using the cosine distance, shown in Figure 4.14, follow similar patterns as the Gaussian selection with KL2 distance (Figure 4.5). Once again, some speakers are favored (SPK4) when using larger amounts of training data. However, when the training data is reduced, the selection algorithm returns balanced proportions of components per speaker.

In order to recreate a more realistic situation in which the amount of training data per speaker is variable, Figure 4.15 shows the proportion of selected components per speaker when the available data per speaker is incremented in blocks of 30s. It can be seen how the proportions of selected components converge to the actual proportions of training data per speaker. However, at low KBM sizes, the selection is insensitive to the actual proportions but, in contrast to the method using KL2 distance, the proportions are less balanced. As an example, it is observed that the proportion of Gaussians for SPK1 is low, regardless on the KBM size.

4.3.2 Cumulative Vectors as speaker models

As shown in Section 4.2.2, BKs are capable of keeping speaker-discriminative information in a vector of binary values. This simple representation enables the use of similarity



(A) 20 min/speaker, 17469 initial components.

(B) 5 min/speaker, 4200 initial components.

(C) 1 min/speaker, 2100 initial components.

(D) 30 sec/speaker, 2100 initial components.

FIGURE 4.14: Percentage of components selected per speaker in the KBM using the cosine distance, with regard to the KBM size, using different amounts of training (20m, 5m, 1m, and 30s per speaker). (F) and (M) indicate male and female speakers, respectively

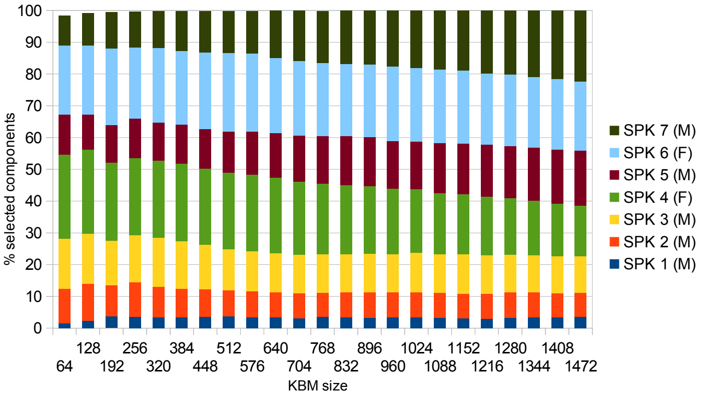


FIGURE 4.15: Percentage of components selected per speaker in the KBM using the cosine distance, with regard to the KBM size, using a different amount of training data for each speaker: 30s, 60s, 90s, 120s, 150s, 180s, and 210s for speakers 1, 2, 3, 4, 5, 6, and 7, respectively.

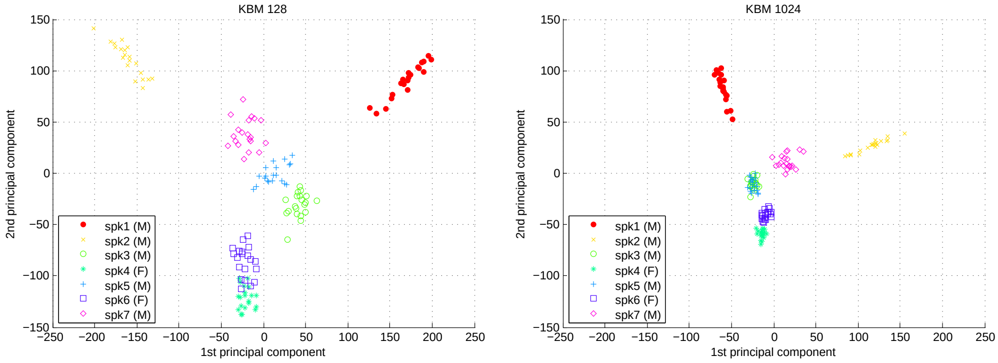


FIGURE 4.16: CVs from several speakers extracted using KBMs of 128 (left) and 1024 (right) Gaussians, projected on the two first directions obtained by PCA.

metrics based on bit-wise operations. An example is the similarity metric defined by Equation 3.2, which has been proven to be effective for the speaker diarization task [Anguera and Bonastre, 2011].

Intuitively, positions equal to one within a BK indicates that those Gaussians of the KBM are the ones that best fit the sequence of feature vectors being converted. Those Gaussians are selected according to the frequency of component activation for the given feature set. In other words, positions in the BK set to one correspond to the top positions in the Cumulative Vector (CV, see Section 3.1.2). The values of the CV can be interpreted as a set of relative weights of each Gaussian of the KBM given the input feature set. Those weights are lost in the conversion of the highest positions to the binary values. Thus, it is reasonable to argue that this missing information could also be helpful for discriminating between speakers. Using CVs instead or in conjunction with BKs has already been addressed for speaker verification with success in [Hernández-Sierra et al., 2012] and [Hernandez-Sierra et al., 2014].

Figure 4.16 shows a set of CVs extracted from the seven speakers of the TTS database using KBMs of 128 and 1024 components, projected on the two principal components obtained after applying PCA. The graphs show quite good separability between speakers even using only the two first principal directions of PCA.

Using CVs as speaker models requires the use of meaningful similarity measures that allows a suitable comparison between speakers. Those similarity measures should ideally emphasize scores between CVs from the same speaker and de-emphasize similarities between CVs from different speakers.

4.3.2.1 Cosine similarity

An effective similarity measure between CVs has to be found. Each position of the CV contains a positive integer representing the count of how many times its associated Gaussian of the KBM has been selected as a top-scoring component. Therefore, the absolute values depend on the length of the set of input features. However, we are more

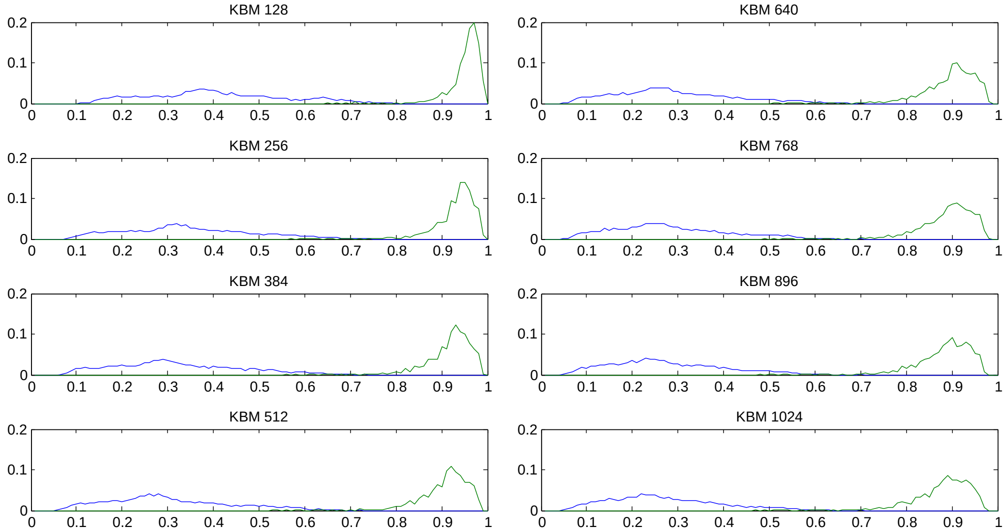


FIGURE 4.17: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **cosine similarity** between CVs calculated on segments of **10s duration**, for different KBM sizes.

interested in the relative weights than in the CVs norm. In other words, we put the focus on the direction of the CVs instead on their magnitudes.

It seems a case where the cosine similarity can be suitable for our purposes. The cosine distance S_{cos}, already defined by Equation 4.8, provides the cosine of the angle between the two vectors. This measure can be seen as a comparison between vectors on a normalized space because the magnitude is not taken into consideration, but the angle between them.

The cosine function returns a real value in the interval $[-1, 1]$. When the directions of two vectors are similar (i.e. the angle between them is near 0 degrees), cosine of angle is near 1, indicating a high similarity. When the two vectors are near orthogonal (angle is near 90 degrees), cosine similarity is near 0. Finally, when the directions of the two vectors are near opposite (angle near 180 degrees), cosine similarity is near -1. As CVs are vectors of positive integers, the angles between CVs are restricted to the interval (0,90) degrees, thus the cosine similarity ranges from 0 to 1.

We can perform a similar analysis of between-speaker and within-speaker distances as the one described in Section 4.2.2 for BKs. Figure 4.17 shows the normalized histograms of within-speaker and between-speaker cosine similarities of CVs estimated on 10s segments. In this case, we can observe how the within-speaker similarities are well concentrated near 1, while the between-speaker similarities are distributed along the remaining range. Both similarity populations are easily separable and a threshold-based assignment could be effectively performed.

It is interesting to check if the more accurate modeling provided by the CVs and cosine similarity is robust against the lack of training data. Figure 4.18 shows the histograms

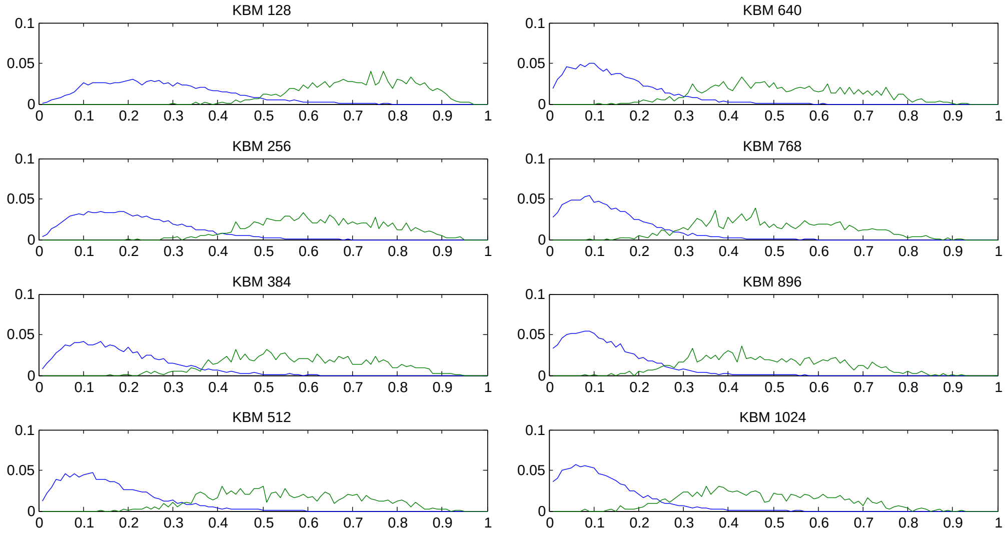


Figure 4.18: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **cosine similarity** between CVs calculated on segments of **1s duration**, for different KBM sizes.

of similarities calculated on segments of 1s duration. As in its BK counterpart, the modeling suffers a loss of accuracy. However, using CVs seems more robust than using BKs, as the overlapping area of between-speaker and within-speaker similarities is lower.

Finally, Figure 4.19 shows the histograms of similarities between CVs estimated on 3s segments. Once again, 3s of speech seem sufficient for estimating good CVs. The histograms show that the two populations are well separated, presenting very few overlapping areas.

If we compare these results with their BK counterparts, everything indicates that the use of CVs to represent speakers and the use of the cosine similarity as metric provide a more accurate speaker modeling, confirming the hypothesis that some speaker-related information is lost in the process of converting a CV into a BK.

4.3.2.2 Chi-squared similarity

Motivated by the success of using CVs as speaker models and the cosine similarity to compare them, we searched for meaningful similarity measures able to exploit the inner nature of the cumulative vectors.

A CV counts how many times each Gaussian component in the KBM has been selected as a top-scoring Gaussian for the feature set being converted. In some way, a CV could be seen as a histogram. Each position of the CV could act as a bin which counts how often its associated Gaussian has been selected. A well-know and commonly used distance metric for comparing histograms is the chi-squared χ^2 distance. This metric takes its name from the Pearson's chi-squared test statistic for comparing discrete probability

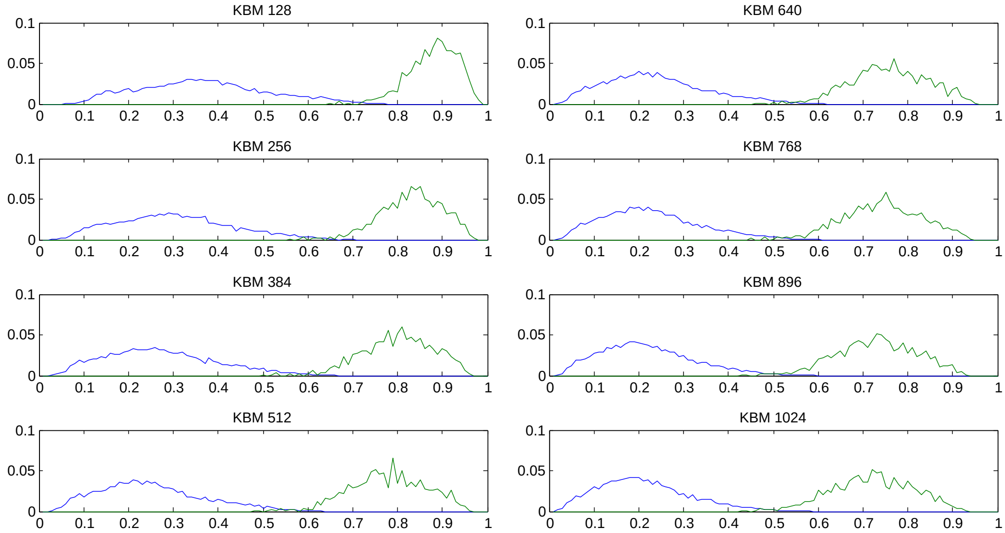


FIGURE 4.19: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **cosine similarity** between CVs calculated on segments of **3s duration**, for different KBM sizes.

distributions. The chi-squared similarity S_{χ^2} is defined as $S_{\chi^2} = 1 - D_{\chi^2}$, where D_{χ^2} is the chi-squared distance defined as

$$D_{\chi^2}(a,b) = \frac{1}{2}\sum_{i=1}^{N}\frac{(a_i - b_i)^2}{a_i + b_i} \tag{4.9}$$

In order to avoid by-zero divisions in the denominator, a constant value (equal to the minimum possible increment defined by double-precision numbers in Matlab, which is 2^{-52}) is summed to all CV positions. Furthermore, the CVs are normalized to have sum equal to 1 before computing the similarity. Proceeding in this way, chi-squared similarity metric ranges in the interval [0,1].

One more time, a similar experiment to the ones performed on the Jaccard and the cosine similarities is conducted, but for the chi-squared similarity. Figures 4.20, 4.21, and 4.22 show the histograms of within-speaker and between-speaker similarities for different KBM sizes and using 10s, 1s, and 3s segments, respectively. The chi-squared similarity exhibits a similar behavior as the cosine similarity. When measuring CVs extracted from 10s segments, the within-speaker and between speaker similarity distributions are well separated. Within-speaker similarities concentrate near 1, while between-speaker similarities are distributed along the similarity range. When using short segments of 1s, similarities get more disperse, but within- and between-speaker similarities are relatively well separated, exhibiting quite little overlapping areas. Once again, similarities between 3s segments seem to provide very good speaker discriminability.

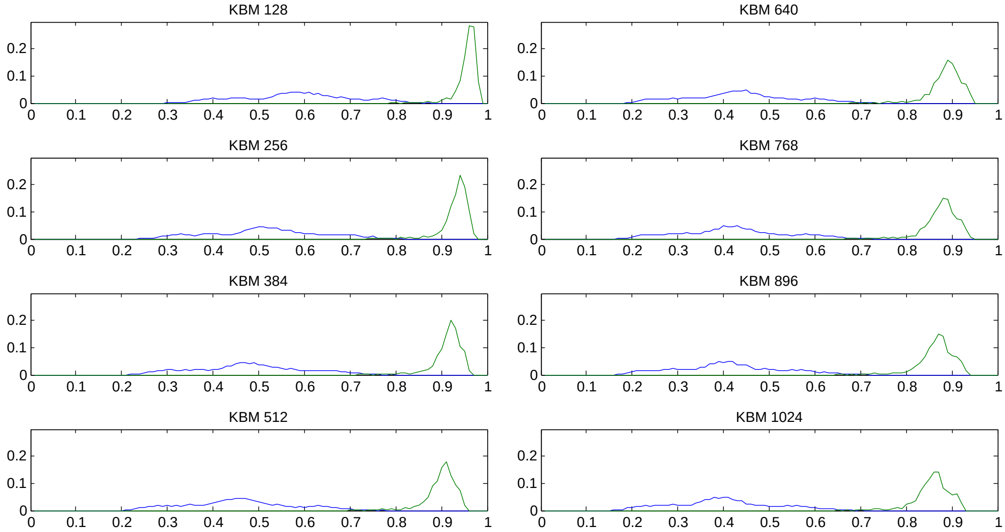


FIGURE 4.20: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **chi-squared similarity** between CVs calculated on segments of **10s duration**, for different KBM sizes.

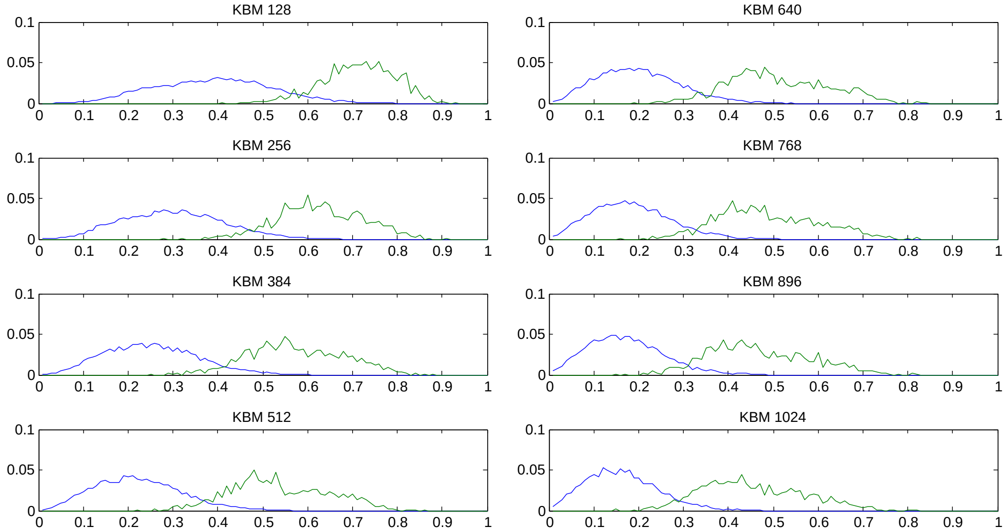


FIGURE 4.21: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **chi-squared similarity** between CVs calculated on segments of **1s duration**, for different KBM sizes.

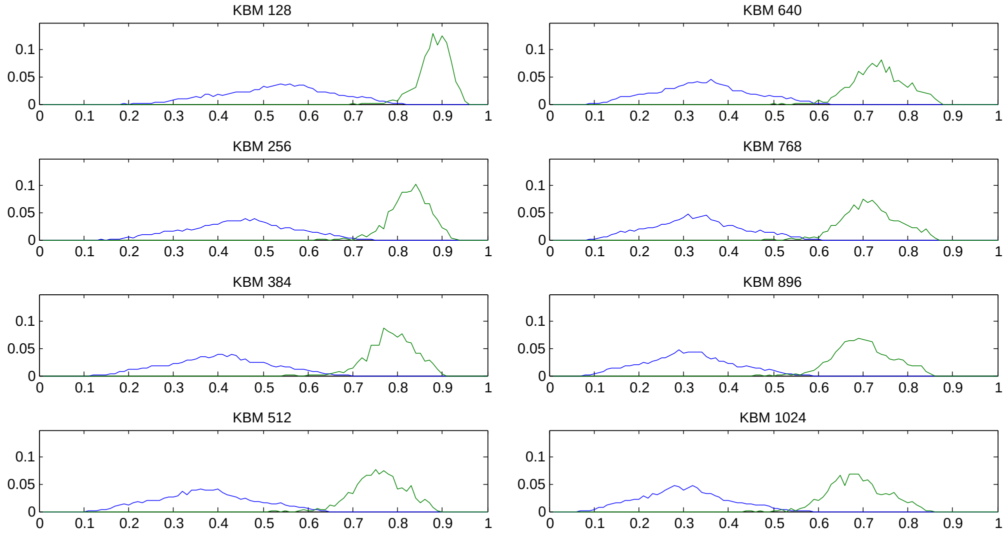


Figure 4.22: Normalized histograms of between-speaker (blue) and within-speaker (green) similarities using the **chi-squared similarity** between CVs calculated on segments of **3s duration**, for different KBM sizes.

4.3.3 Final clustering selection

One of the key points in any Agglomerative Hierarchical Clustering (AHC) is to determine when to stop the process at the right number of clusters. This is usually referred to as stopping criterion. Commonly, each partial clustering solution obtained at each AHC iteration is evaluated in order to decide whether the current solution is a good data partition or if it is necessary to continue the cluster merging. Commonly in GMM-based speaker diarization systems, ΔBIC is calculated for all cluster pairs at the current iteration. If $\Delta\text{BIC} < 0$ for all cluster pairs, then the AHC is stopped and the current clustering solution is returned as the output.

A second option is to continue the AHC process until the end, i.e. until all the data is grouped into a single cluster. All the partial clusterings obtained at every iteration are stored, and it is at the end when the optimum clustering solution is selected from those partial solutions generated. In this case there is no a stopping criterion, but a final clustering selection stage. The optimum clustering could be selected with regard to a threshold previously calculated on development data. However, the dependence on development data may cause robustness issues when the nature of the testing data is unknown.

An alternative family of clustering selection techniques is based on the relations between within-cluster and between-cluster distances of the elements within a clustering solution. A good clustering solution will ideally present a good separation between within-class and between-class distance populations. These techniques are presented as an alternative to other techniques which require parameter tuning on development data.

The final clustering selection technique used in our baseline binary key speaker diarization system follows this principle: the within-speaker and between-speaker similarity populations are modeled by single Gaussians, and then the student T-test statistic is computed for every clustering solution. Finally, the clustering which maximizes T-test is selected as the optimum solution.

However, it has been reported in [Anguera and Bonastre, 2011] under a meeting recordings environment, and in [Delgado et al., 2014b] under a TV broadcast data environment, that this clustering selection algorithm is far from returning clustering solutions near to the optimum ones. In fact, this point has been identified as the main weakness of the complete system because it has an important negative impact on system performance. In Section 3.3.3 the clustering selection stage was evaluated and the obtained DER results are well above the DER floor. It is for this reason that we believe that an effective clustering selection would systematically result in an increase of performance. It is a must to propose a different approach that can get closer to the performance ceiling.

In [Delgado et al., 2015c], we proposed a clustering selection technique based on the Within-Cluster Sum of Squares (WCSS). Given a clustering solution C_k composed of k clusters $c_1, c_2, ..., c_k$, the WCSS, $W(C_k)$, is defined as

$$W(C_k) = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - \mu_i\|^2 \tag{4.10}$$

where μ_i is the mean of the points of cluster c_i (i.e. the centroid of cluster c_i). Actually, WCSS is the objective function used by the k-means clustering algorithm. But the WCSS can also be used as an indicator of how good a clustering solution is. Presumably, a good clustering solution originates clusters with small WCSS. However, it must be taken into account that the clustering solution with minimum WCSS is the one in which each point becomes a single cluster. However, this is not usually the required solution. On the contrary, the clustering with the highest WCSS is the one in which all the data is grouped in a single cluster. It is therefore necessary to find a trade-off between WCSS and the number of clusters.

Given a set of clustering solutions $C = (C_1, ..., C_{N_{init}})$, each one with an increasing number of clusters (from a single cluster to N_{init} clusters), WCSS can be calculated for all clustering solutions and plotted as shown in Figure 4.23. When the number of clusters N is less than the optimum number of clusters, WCSS should be high. In the case of $N = 1$, WCSS is maximum, and when increasing the number of clusters, WCSS will exhibit an exponential decay. In some point the decay will show almost linear behavior and WCSS will continue to fall smoothly. The first point which deviates from the exponential model is considered the elbow, and the associated number of clusters is selected as the optimum one.

The elbow cannot always be unambiguously determined. We propose a simple graphical approach to find the elbow, which consists in drawing a straight line between the WCSS values of the first (with $N = 1$) and last ($N = N_{init}$) clustering solutions and calculate the distance between all points in the curve to the straight line. Then, the point with the longest distance to the straight line is selected, and the number of cluster will be the X value of such point.

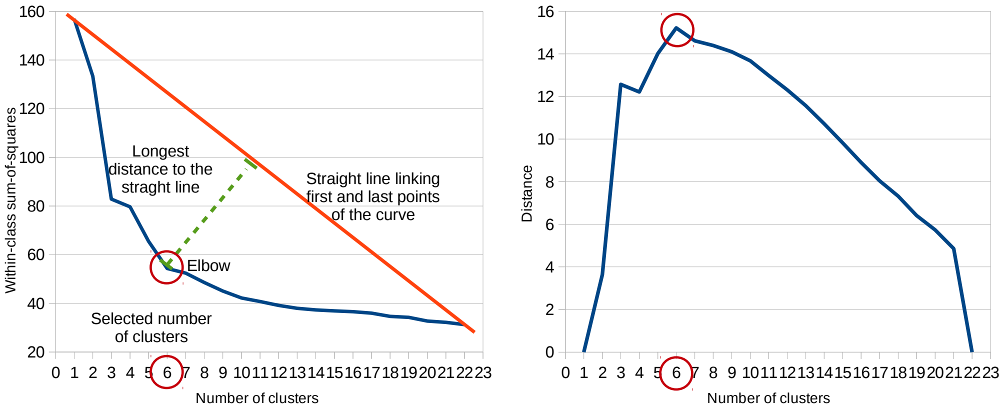


FIGURE 4.23: Example of the elbow criterion applied over the curve of within-class sum-of-squares per number of clusters (left). The point with longest distance to the straight line is considered the elbow (right).

In Equation 4.10, the Euclidean distance of each cluster member to its centroid is used. However, as we are using CVs which represent speech segments, we find more adequate to use one of the studied distance metrics for CVs. Therefore, the Euclidean distance may be replaced by one of the proposed distance metrics for the comparison of CVs. WCSS for a clustering solution C_k of k clusters, $W(C_k)$, can be reformulated as

$$W(C_k) = \sum_{i=1}^{k} \sum_{x \in c_i} (D(x, \mu_i))^2 \tag{4.11}$$

where D is a suitable distance measure between CVs, e.g. the cosine distance, or the chi-squared distance.

4.3.4 Intra-session variability compensation

It is well known how the changing acoustic conditions affect negatively to speaker modeling approaches which usually perform adequately on ideal conditions. During the last years much effort has been put on the compensation of such channel effects. Today, dealing with session variability has become a must for any modern speaker recognition system. The last achievements in this regard have resulted in impressive speaker recognition accuracy improvements. Chronologically, some of these session compensation methods are Nuisance Attribute Projection (NAP) [Solomonoff et al., 2004]), Within-Class Covariance Normalization (WCCN) [Hatch et al., 2006], Joint Factor Analysis (JFA) [Kenny et al., 2008]) and total variability, also known as the i-vector paradigm [Dehak et al., 2011]. Current state-of-the-art performance in speaker verification is provided by the i-vector paradigm combined with Probabilistic Linear Discriminant Analysis (PLDA) [Garcia-Romero and Espy-Wilson, 2011].

In speaker recognition, channel variability is usually present, for example, when using different microphones for speaker enrollment and testing, or because the background acoustic conditions may change between enrollment and testing. This variability is referred to as inter-session variability.

In a speaker diarization task, there is no actual inter-session variability since there is only one session involved (in the case of classical, single-session speaker diarization). However, the acoustic conditions can vary within a session. For instance, in a broadcast news program, the anchor speaker can speak over jingles, background noise, or in a silent environment. We refer to those situations as intra-session variability. One of the consequences of intra-session variability is that a given speaker could be modeled by more than one cluster, each one for a different channel condition. Ideally, all those clusters referring to a same speaker should be merged together to form a single speaker cluster. It is reasonable to think that reducing the impact of channel variability could be helpful to agglomerate all segments from a single speaker together, even under changing channel conditions.

As explained in Chapter 2, some of the most recent advanced approaches to speaker recognition have already been successfully applied to speaker diarization. Examples of such works are [Aronowitz, 2010], which uses NAP over Gaussian supervectors, [Castaldo et al., 2008], which uses JFA modeling, and [Silovsky and Prazak, 2012] and [Rouvier et al., 2013], which use i-vectors.

The binary key speaker modeling is not an exception, and some work has been done to perform session variability compensation. In [Hernandez-Sierra et al., 2014], some popular methods are adapted and applied successfully in a speaker verification framework. These methods are NAP, WCCN and some variants of them.

In [Delgado et al., 2015b], we proposed to introduce NAP into our binary key speaker diarization system. Since NAP applied to the cumulative vector space has been shown to be successful in a speaker verification task, the aim of that paper was to assess if it would also be effective in speaker diarization. Next, a brief review of NAP method is done and its adaptation to our binary key speaker diarization system is described.

Nuisance Attribute Projection

Given a set of speech utterances, each one represented by a supervector, NAP [Solomonoff et al., 2004] assumes that the within-class variability is restricted to a low dimensional subspace. In order to remove this variability, the supervectors are projected onto an orthogonal complementary subspace. First, the within-speaker scatter matrix is calculated on appropriate labeled data as

$$\mathrm{W} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}(z_i^s - \bar{z}^s)(z_i^s - \bar{z}^s)^t \qquad (4.12)$$

where S is the number of speakers, n_s is the number of utterances by speaker s, z_i^s is the supervector representing the i-th utterance of speaker s, and $\bar{z}^s$ is the mean of the supervectors of speaker s. Then, the projection matrix is obtained according to

$$P = (I - UU^t) \tag{4.13}$$

where U is the rectangular matrix of the k eigenvectors associated with the k largest eigenvalues (obtained after solving the eigenvalue problem $Wu = \lambda u$). Finally, the transformation of supervector x is the result of applying the projection as

$$y = Px \tag{4.14}$$

Given its nature, NAP can be applied to any sort of supervector representation of an utterance/cluster. Within our speaker diarization scheme, recall that the input feature stream is converted into a sequence of binary keys or cumulative vectors. It is at this point where the obtained sequence of CVs representing the input data are compensated by applying the projection defined in Equation 4.14. Next, in the AHC stage, the compensated CVs are assigned to the current clusters, and cluster CVs are retrained. After those retraining steps, cluster CVs are also compensated through Equation 4.14. The rest of the process remains the same as the baseline system.

With regard to the estimation of the within-class scatter matrix W, a development set is required. As our system estimates the KBM on the test audio file, the CVs of the development set have to be calculated for each test audio file using its own KBM.

Once matrix W is estimated, the projection is calculated by means of Equation 4.13. As said above, U is the matrix formed by the k eigenvectors associated to the top k eigenvalues. The number k of eigenvectors is a parameter that needs to be tuned.

4.4 Evaluation

In this section, we describe the experiments carried out in order to assess our proposals to improve the baseline single-session speaker diarization system based on binary keys. These proposals under evaluation are the ones described in Section 4.3. First, the modifications concerning the KBM training, the use of CVs as speaker models, and the selection of the final clustering are evaluated in Section 4.4.1. Then, the evaluation of intra-session variability compensation is provided separately in Section 4.4.2.

As in the evaluation of the baseline system described in Section 3.3, we use the REPERE phase 1 dataset. In this case, first all the modifications are evaluated on the REPERE phase 1 test set. Once all the proposals are evaluated, the best system found is also evaluated on the REPERE phase 1 development set, with the aim of checking if performance is consistent when using a different unseen dataset.

The used evaluation metric is the Diarization Error Rate (DER), calculated by following the method proposed by LNE (see Section 2.4.1 for further details).

4.4.1 Single-session speaker diarization experiments

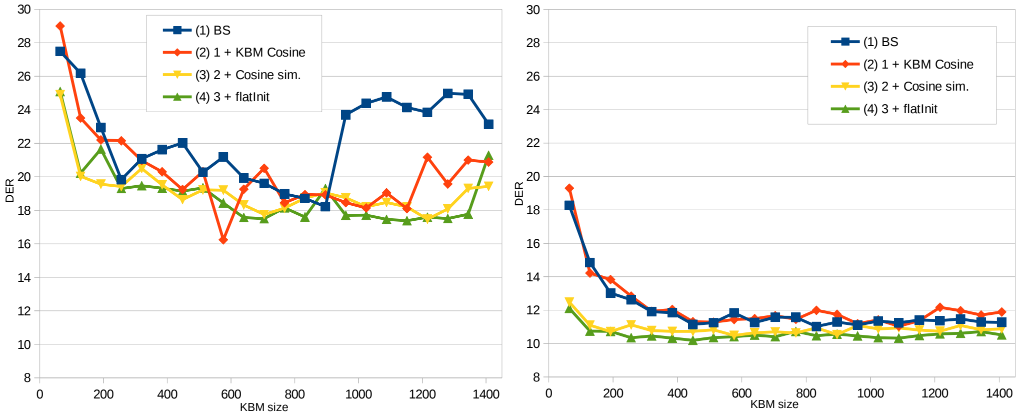
The basic settings of the single-session speaker diarization experiments are essentially the same used for the evaluation of the baseline system. System settings are summarized below:

- **Feature extraction**: 19-order MFCCs are extracted from the audio stream using a 25ms window every 10ms.
- **KBM training**: Single, diagonal-covariance Gaussian models are trained on the input sequence of MFCCs using a sliding window of 2s and a window shift dependent on the duration of the input data. The value of the window shift is set in order to obtain a pool of Gaussian of not less than 2000, and cannot exceed 0.5s.
- **Binary key estimation**: At feature level, the top 5 components are first selected. Then, at utterance/cluster level, the top 20% positions of the CV are set to 1 and the rest are set to 0.
- **Initial number of clusters**: The initial number of clusters N_{init} is set to 25 for all sessions. This value is selected in order to assure an initial number of clusters some units greater than the actual number of speaker, which can be up to 18.
- **Clustering initialization**: The initial clusters are initialized using small segments of 100ms, assigning them to the clusters through maximum likelihood. The initial cluster models are the first N_{init} components of the KBM.
- **Input data BK extraction**: The input sequence of features is divided into 1s segments, extending them 1s after and before, totaling 3s. Then, a BK is extracted from each segment.

In a first block, we evaluate some of the proposed modifications in an incremental basis: each result is obtained from the previous system plus the addition of a new improvement, as described below:

- (1) Baseline system.
- (2) 1 adding KBM training improvement.
- (3) 2 adding use of cosine similarity and CVs.
- (4) 3 adding flat cluster initialization.

Figure 4.24 shows system performance on the REPERE test dataset when adding the modifications according to the list above. In addition, for all the given system configurations, we provide DER of the labels returned by the system by using the best clustering selection algorithm (Figure 4.24a), and DER of the optimum labels which the system is able to generate, i.e. all partial clusterings generated by the system are evaluated, ignoring the best clustering selection, and then the best labeling in terms of DER is selected manually (Figure 4.24b). These last measurements are provided in



(A) System output.

(B) Optimum output.

FIGURE 4.24: Diarization performance after applying the proposed improvements one by one: DER of system output (Figure 4.24a) and DER of optimum output (Figure 4.24b).

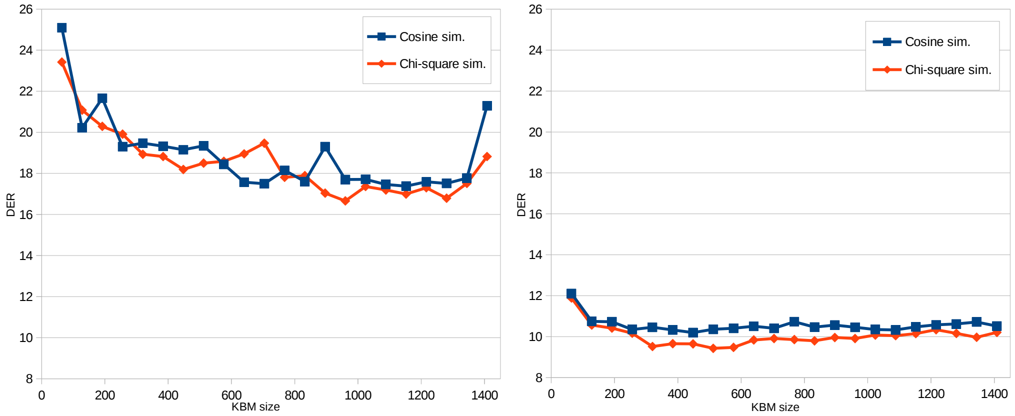
order to establish a performance ceiling which enable us to check how close the clustering selection algorithm output is to the optimum clustering. All the results are plotted as a function of the KBM size (X axis).

The first result shown (1) is the baseline performance, corresponding to the same figures shown in Figure 3.6. (2) corresponds to the baseline system but replacing the KBM training procedure by the one described in Section 4.3.1. We observe an improvement of the system output (Figure 4.24a), particularly for KBM sizes greater than 896. As for the performance ceiling (Figure 4.24b), both KBM training methods perform similarly. Even so, as it will be shown later, the best improvement of this modification is undoubtedly in terms of computation time.

The next improvement consists in using the cosine similarity between CVs (3). Generally, performance gets even better in both system output and performance ceiling. As it was hypothesized, the information contained in the CVs helps to better discriminate between speakers.

The next result corresponds to replacing the cluster initialization strategy by a straightforward uniform clustering initialization consisting in dividing the input signal into N_{init} equal-sized chunks which define the initial set of clusters (4). Surprisingly, using such a rough uniform initialization performs better than the method explained in Section 3.2.1.3. This indicates that the AHC scheme is able to reallocate the data from the initial uniform clusters towards actual speaker clusters.

After applying this series of modifications, improvements in performance have been achieved. However, the gap between system output and performance ceiling is still large. The best system output is not smaller than 16% DER, whilst the error floor is slightly above 10%.



(A) System output.

(B) Optimum output.

FIGURE 4.25: System performance by using the cosine and chi-squared similarity measures: DER of system output (Figure 4.24a) and DER of optimum output (Figure 4.24b).

In a second block, we aim at comparing the cosine and chi-square similarities for CVs. For this end, we use the new KBM training algorithm and the flat clustering initialization method (system 4 of previous experiment). Then, the experiment is performed for both cosine and chi-squared similarity metrics.

Figure 4.25 shows the obtained results for both system output (Figure 4.25a) and performance ceiling (Figure 4.25b). With regard to the system output, both similarity metrics yield similar performance. However, it seems that the system using the chi-squared similarity presents a slightly better performance for the majority of tested KBM sizes. As for the performance ceiling, chi-squared similarity also shows a slight improvement in performance.

A third block of experiments aims at evaluating the newly proposed algorithm for the selection of the best clustering solution. We select the two systems described in the second block of experiments (using the cosine and the chi-squared similarities, respectively), and we replace the original final clustering selection technique by the one based on the within-cluster sum of squares proposed in Section 4.3.3. This technique requires the use of a distance metric as well, thus we opt for evaluating both the cosine and the chi-squared distances. As a result, 4 different combinations are obtained according to the similarity/distance metric used in the AHC process and in the final clustering selection:

- COS-COS: cosine similarity in AHC / cosine distance in clustering selection.
- COS-CHI: cosine similarity in AHC / chi distance in clustering selection.
- CHI-COS: chi similarity in AHC / cosine distance in clustering selection.
- CHI-CHI: chi similarity in AHC / chi distance in clustering selection.

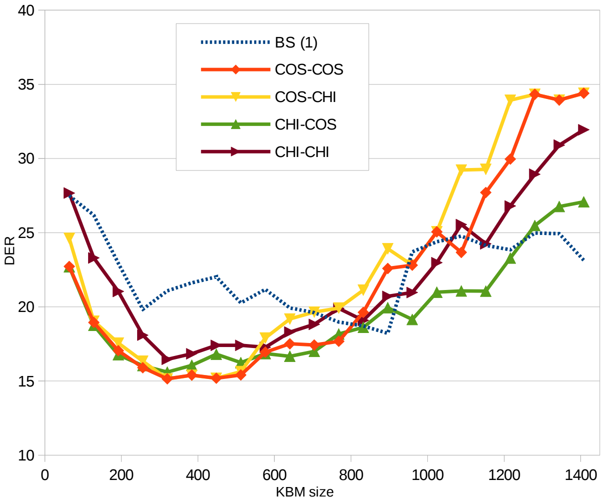


Figure 4.26: System performance after applying the new clustering selection algorithm, for the 4 possible combinations of cosine and chi-squared similarities/distances used in AHC and clustering selection.

Figure 4.26 depicts the results obtained for the 4 possible combinations. The baseline performance is also provided for comparison ("BS (1)"). It can be observed how the four combinations outperform the baseline system, and present a similar pattern with regard to the size of the KBM: the best performances are obtained for KBM sizes between 256 and 576, getting DER values near 15%. Then, performance increases with the size of the KBM. With regard to the similarity metric used in the AHC process, the 2 combinations using the cosine similarity are the best performing ones, against the ones using the chi-squared similarity. This result may seem contradictory, as in Figure 4.25b the obtained performance ceiling using the chi-square similarity is slightly better than the one obtained with the cosine similarity. One could argue that the clustering selection algorithm could work better on the set of clustering solutions produced by that system. However, this is not the case. Anyway, the best obtained performances are quite similar for the 4 combinations, regardless on the similarity metric used in the AHC. As for the distance metric used in the selection technique, once again the cosine distance exhibits a slightly better behavior than the chi-square distance. However the differences in performance are just subtle.

One remarkable fact is the different trend of DER of the system output and the performance ceiling when incrementing the KBM size. It can clearly be seen that the performance ceiling (Figure 4.25b) converges at around 10% DER, while performance obtained by the clustering selection suffers an important degradation for bigger KBMs. In order to find a reason to this fact, Figure 4.27 shows the WCSS (Y axis) curves calculated on the set of clustering solutions (X axis) of a given audio file, for different sizes of the KBM. As it can be observed, WCSS becomes higher when increasing the KBM size. This is a direct consequence of a global increase of the distance population when the KBM becomes bigger. This fact was already shown in Figure 4.19 which shows the histograms of the within-cluster and between-cluster cosine similarities for different

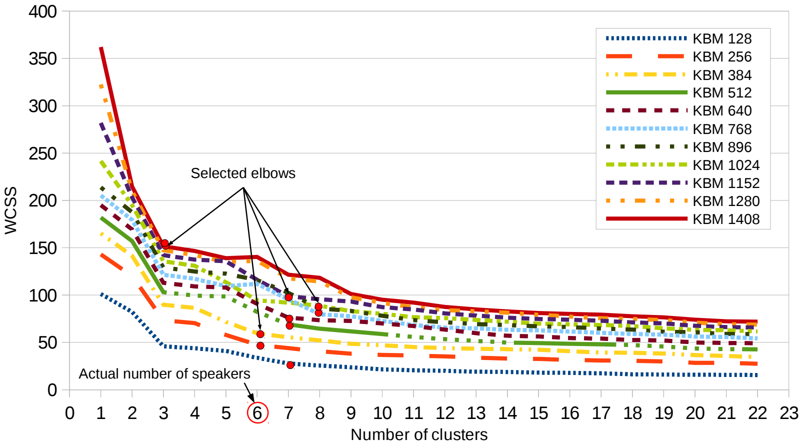


FIGURE 4.27: Example of the elbow criterion over the WCSS curve for a particular audio session, calculated for different KBM sizes. The red points indicate the selected number of clusters.

KBM sizes. With an increasing size of the KBM, the similarities population tends to move to the left part of the histogram, indicating that the similarities tend to be smaller (and consequently, distances tend to be larger). Therefore, WCSS also gets larger with an increasing number of KBM components. The final consequence of this is that the slope of the exponential part of the WCSS curve becomes more and more pronounced, and this results in a tendency to over-cluster the data, producing less clusters than the actual number of speaker (see selected clusters and actual clusters in Figure 4.27), which explains the sudden DER increase when the KBM size becomes higher. Oppositely, when the KBM size is small, the algorithm tends to under-cluster, which also explains the higher values of DER for those KBM sizes.

After these three blocks of experiments, we can identify the best performing system. This is the one using the new KBM training approach, the cosine similarity in the AHC process, the uniform clustering initialization, and the new clustering selection algorithm with the cosine distance. The best performance achieved is 15.15% DER, obtained using a KBM size of 320. This supposes a 16.8% relative improvement with regard to the best performing configuration of the baseline system (18.22% DER). Table 4.4 provides results broken-down into the different sessions. In a first look to system performance ("DER (#S)" column) and performance ceiling ("DER floor (#S)") one can see how the final clustering selection technique is able to return the optimum clusterings for some audio sessions (BFMStory_1, BFMStory_4, CaVousRegarde_1, CaVousRegarde_3, EntreLesLignes_2, PileEtFace_2, PileEtFace_3, PileEtFace_5, TopQuestions_1, and TopQuestions_3). As in the case of the baseline system evaluated in Section 3.3.3, *Culture et vous* sessions show quite high error rates, due to the short audio duration and the high number of speakers per session. In fact, the number of clusters of the automatically selected clustering solutions for those files is much lower than the actual number of speaker. Regarding false alarm errors, these error rates should be zero since ground-truth labels are used for SAD. However, as we explained in Section 3.3.3 (in which the baseline system

Session ID (#S)	Speech time	**FA**	**MISS**	**SPKE**	**DER (#S)**	**DER floor (#S)**
BFMStory_1 (6)	14:43.67	0.13	4.79	1.34	6.26 (6)	6.26 (6)
BFMStory_2 (18)	13:43.50	0.55	0.33	25.00	25.88 (9)	17.14 (12)
BFMStory_3 (10)	14:23.70	0.28	1.00	10.30	11.58 (6)	8.37 (9)
BFMStory_4 (6)	14:41.29	0.06	2.65	1.53	4.24 (6)	4.24 (6)
CultureEtVous_1 (5)	01:42.39	1.16	0.06	25.32	26.54 (4)	24.49 (11)
CultureEtVous_2 (6)	03:48.76	0.85	1.33	39.94	42.12 (4)	31.82 (6)
CultureEtVous_3 (16)	01:35.04	0.81	1.90	37.58	40.28 (3)	20.70 (8)
CultureEtVous_4 (9)	01:44.11	0.01	0.59	27.38	27.98 (3)	6.84 (7)
CultureEtVous_5 (6)	01:26.38	0.03	3.53	15.65	19.22 (2)	13.34 (7)
CultureEtVous_6 (12)	01:41.06	0.78	0.98	26.83	28.59 (3)	22.44 (12)
CultureEtVous_7 (14)	01:28.11	2.22	2.01	56.64	60.87 (3)	24.96 (14)
CaVousRegarde_1 (7)	04:43.98	0.85	0.88	6.78	8.51 (4)	8.51 (4)
CaVousRegarde_2 (5)	04:52.29	0.48	4.41	10.23	15.12 (6)	7.64 (5)
CaVousRegarde_3 (5)	05:01.55	0.49	9.25	13.28	23.01 (6)	23.01 (6)
EntreLesLignes_1 (5)	04:47.95	0.98	5.89	30.81	37.68 (7)	29.33 (11)
EntreLesLignes_2 (5)	04:55.59	0.63	3.24	7.30	11.17 (6)	11.17 (6)
EntreLesLignes_3 (5)	04:55.20	0.32	0.96	4.53	5.80 (5)	5.80 (5)
LCPInfo13h30_1 (16)	09:46.55	0.25	0.49	13.47	14.22 (7)	9.87 (9)
LCPInfo13h30_2 (12)	09:34.18	0.05	0.06	25.42	25.53 (6)	2.34 (12)
LCPInfo13h30_3 (10)	09:42.73	0.08	1.33	14.48	15.89 (5)	11.74 (11)
PileEtFace_1 (3)	02:59.67	0.83	6.02	18.67	25.51 (5)	15.94 (3)
PileEtFace_2 (3)	02:58.06	0.66	5.98	5.57	12.21 (4)	12.21 (4)
PileEtFace_3 (3)	02:58.96	0.68	7.12	22.68	30.49 (6)	30.49 (6)
PileEtFace_4 (3)	02:55.03	0.40	4.48	15.57	20.45 (2)	5.22 (4)
PileEtFace_5 (3)	02:53.46	0.35	3.38	1.53	5.26 (3)	5.26 (3)
TopQuestions_1 (8)	10:41.41	0.15	0.95	2.44	3.55 (8)	3.55 (8)
TopQuestions_2 (5)	06:39.55	0.01	0.56	0.25	0.82 (5)	0.82 (6)
TopQuestions_3 (6)	07:26.76	0.49	0.18	3.48	4.15 (5)	4.15 (5)
Overall (time-weighted)	-	0.37	2.35	12.43	15.15 (-)	10.46 (-)

Table 4.4: Performance measures in a per-file basis of the best performing system, for a KBM size of 320, broken down into false alarm (FA), miss error (MISS), and speaker error (SPKE). DER of the best clustering outputs selected manually are also shown (DER floor). (#S) indicates the number of speakers/clusters.

was evaluated), false alarm rates calculated by the scoring tool by LNE (optimized for the evaluation of overlapping speech), are affected by the way the forgiveness collar is applied. The same phenomenon occurs in the computation of miss speaker error rates (which should ideally be equal to the overlapping speaker time), which are also affected by the method used in the forgiveness collar application. These effects make rather difficult to get a meaningful interpretation of both false alarm and miss speech error rates.

Up to now, the new improvements have been evaluated in terms of performance. However, we have to keep in mind that our priority is to get a very fast speaker diarization system useful for processing large collections of data in a reasonable time. Figure 4.28 show execution times in terms of real-time factor (xRT) as a function of the KBM size, after applying the proposed improvements one by one. (1) corresponds to the baseline execution time, (2) includes the new KBM training method, (3) introduces the cosine similarity, (4) includes the uniform cluster initialization, and finally (5) uses the new clustering selection. We observe an impressive decrease in execution time for (2). In fact, such modification was proposed specially for accelerating the KBM training, but it has also been shown that the new KBM play its role as effectively as its predecessor. The rest of improvements, although not intended for improving speed, have also a positive impact on execution time. The use of the cosine similarity (2) in place of the Jaccard similarity has resulted in a slight decrease in execution time, although this improvement

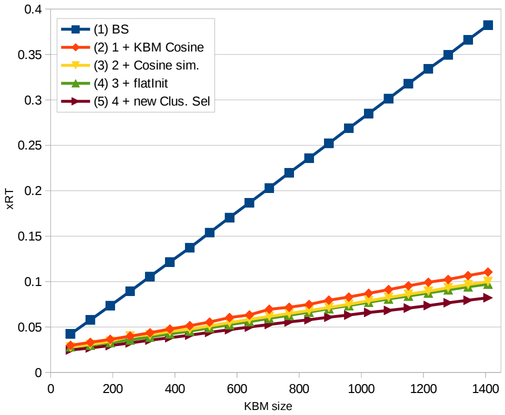


FIGURE 4.28: Execution time measured in xRT, with regard to the KBM size, for the baseline system, and after applying the proposed improvements one by one.

may be due to implementation matters (the use of optimized, built-in Matlab functions for the cosine calculation). The flat clustering initialization (4) has also resulted in speed gains, as this method has no cost at all. Finally, the new clustering selection technique (5) is also more efficient than the original one, as it only requires the computations of within-class distances, while the original method requires the calculation of both the within- and between-cluster distances.

The previous best-performing system selected (with KBM size of 320) presents an average execution time of 0.0354 xRT (around 28 times faster-than-real-time). The best performing baseline system presented a 0.252 xRT (around 4 times-faster-than-real-time). Therefore, the new system is around 7 times faster than the baseline.

In Section 3.3.3, an experiment was carried out to investigate the effect of the audio duration in the training of the KBM. There, we observed important differences in execution time, especially between short and long audio files. Here we take the same audio files and repeat the measurements using the new best-performing system. Results are collected by Table 4.5. It is observed that xRTs are now significantly lower. One of the reasons is that the new more accurate modeling allows us to use smaller KBMs, resulting in speed ups in all the stages involved in diarization. But the main reason is that the component selection algorithm of the KBM training is now much faster. Still we can observe a difference between short and long audio files (in this example, both files require practically the same time for KBM estimation), but this difference is drastically reduced and does not have a great impact in the calculation of the overall xRT for the whole dataset.

Finally, the last point is to verify if the best obtained system performs similarly on a different data set. Figure 4.29 shows system DER on the REPERE test ("REP. TEST SysOut" and "REP. TEST OptOut") and development ("REP. DEV SysOut" and "REP. DEV OptOut") datasets together. Execution time measured in xRT is also provided. We observe that the system behaves very similarly with the two datasets, but with an approximately constant decrease in performance of around 2% absolute when processing

	Audio file 14:43 (883s)		Audio file 01:28 (88s)	
Stage	Time (s)	xRT	Time (s)	xRT
KBM training	**1.07**	**0.0012**	**1.39**	**0.0158**
BK estimation	10.59	0.0119	1.43	0.0163
Clustering init.	0.0004	0.0000	0.0003	0.0000
AHC	13.83	0.0156	2.34	0.0266
Clustering selection	0.60	0.0006	0.21	0.0024
Overall	26.09	**0.0293**	5.37	**0.0611**

Table 4.5: Execution time (with KBM size = 320) measured in seconds and xRT for two audio files of different durations (14:43 and 01:28, respectively) after applying speeding improvements, broken down into the different stages of the diarization process

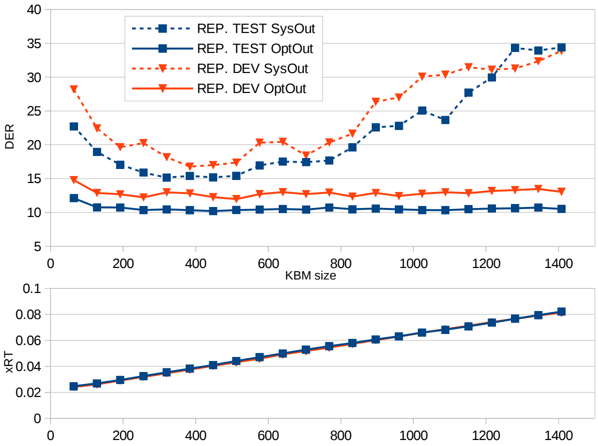


Figure 4.29: Comparison of performance of the best performing system when processing the REPERE test and development sets: DER of system output, DER of optimum output, and xRT.

the REPERE development set. As far as execution time concerns, xRT are practically identical. The best operating point of the system on the new dataset is the same as for the REPERE test set: with a KBM of 320 components, DER is 18.17% (just 2% absolute above).

4.4.2 Intra-session variability compensation evaluation

In this subsection we evaluate the effectiveness of the Nuisance Attribute Projection applied to our binary key based speaker diarization system, as it has been explained in Section 4.3.4. We take the best performing system found in the previous subsection (with the new improvements and a KBM of 320 components) and then perform the variability compensation on the input and cluster CVs.

Estimating the within-class scatter matrix requires development data. In this experiment, we use the REPERE phase 1 development set to estimate W. For each audio file in the development set, all the segments of each participating speaker are pooled

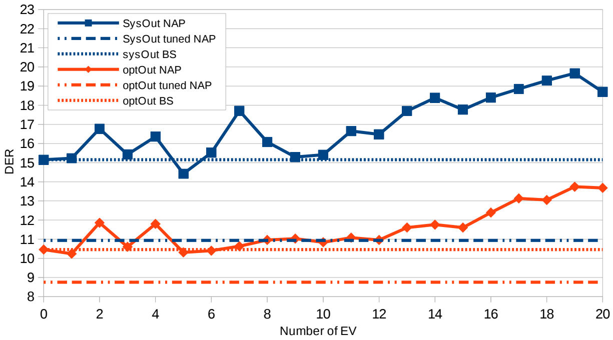


Figure 4.30: Diarization performance after applying NAP compensation, as a function of the number of eigenvectors selected. Results are calculated for both system and optimum clusterings.

together by speaker, and divided into segments of one second. Those segments are used as speaker utterances in the computation of W. Note that prior to the calculation of W, the development set has to be converted into a sequence of CVs. Given that each test file is processed with its own KBM, the development set CVs have also to be estimated using the KBM for the current test file. This implies that the development CVs cannot be shared session-wise: the development CVs have to be recalculated for each test file using its own KBM.

Figure 4.30 shows DER as a function of the number k of eigenvectors used for the estimation of NAP. As usual, system output ("SysOut NAP") and optimum output ("optOut NAP") performances are evaluated. The baseline results ("SysOut BS" and "optOut BS") are also included as a reference. It can be seen how performance is not very stable across k. In the system output, only $k = 5$ outperforms the baseline, providing a slight improvement of less than 1% absolute DER. In the optimum outputs, DER seems to be more stable, but again the improvement is very small.

As each session is processed using its own KBM trained on the session itself, it could be argued that a fixed value of k may not be the best option. For this reason, we show in Table 4.6 the results of applying NAP in a per-session basis by tuning k for each session. It is shown how the optimal k is totally dependent on the audio session. In this case, practically all sessions benefit from the application of NAP, obtaining significant error reductions. For the system output, an overall DER of 10.71% is obtained against the 15.15% of the baseline. For the optimum output, overall DER obtained is 8.75% against 10.46% of the baseline system. These results indicate that the use of NAP for intra-session variability compensation is meaningful. However, selecting the right number k of eigenvectors has still to be solved.

In this regard, we could try to select k adaptively on the current session. Instead of using a fixed value of k for all input audio files, we could estimate k as a function of the variance explained by each eigenvector, i.e. as a function of the eigenvalues associated to the eigenvectors. Particularly, k could be selected as a function of the proportion p of

Session ID	SysOut			OptOut		
	w/o NAP	NAP	#EV	w/o NAP	NAP	#EV
BFMTV_BFMStory_1	6.26	6.06	15	6.26	5.57	15
BFMTV_BFMStory_2	25.88	17.93	6	17.14	15.88	19
BFMTV_BFMStory_3	11.58	3.32	20	8.37	3.32	13
BFMTV_BFMStory_4	4.24	4.34	9	4.24	4.34	20
BFMTV_CultureEtVous_1	26.54	24.59	1	24.49	13.48	1
BFMTV_CultureEtVous_2	42.12	32.26	1	31.82	31.15	12
BFMTV_CultureEtVous_3	40.28	31.17	4	20.7	19.73	12
BFMTV_CultureEtVous_4	27.98	14.79	1	6.84	2.55	1
BFMTV_CultureEtVous_5	19.22	19.22	6	13.34	9.85	1
BFMTV_CultureEtVous_6	28.59	34.36	4	22.44	18.21	12
BFMTV_CultureEtVous_7	60.87	60.87	3	24.96	23.04	1
LCP_CaVousRegarde_1	8.51	8.58	7	8.51	8.37	2
LCP_CaVousRegarde_2	15.12	7.66	14	7.64	7.66	6
LCP_CaVousRegarde_3	23.01	20.56	6	23.01	19.29	12
LCP_EntreLesLignes_1	37.68	33.21	14	29.33	28.97	1
LCP_EntreLesLignes_2	11.17	11.36	5	11.17	11.36	16
LCP_EntreLesLignes_3	14.22	4.91	11	5.8	4.91	3
LCP_LCPInfo13h30_1	25.53	8.72	13	9.87	6.34	6
LCP_LCPInfo13h30_2	15.89	3.09	6	2.34	0.31	20
LCP_LCPInfo13h30_3	25.51	12.99	2	11.74	7.1	8
LCP_PileEtFace_1	12.21	16.01	3	15.94	16.01	14
LCP_PileEtFace_2	30.49	12.79	3	12.21	12.79	14
LCP_PileEtFace_3	20.45	27.46	15	30.49	27.46	7
LCP_PileEtFace_4	5.26	3.55	5	5.22	3.55	16
LCP_PileEtFace_5	3.55	5.76	9	5.26	5.76	20
LCP_TopQuestions_1	0.82	3.41	11	3.55	2.7	15
LCP_TopQuestions_2	4.15	0.68	15	0.82	0.68	7
LCP_TopQuestions_3	5.8	3.93	1	4.15	3.93	1
Overall	15.15	10.71	-	10.46	8.75	-

Table 4.6: Diarization performance broken down into audio sessions, before and after applying NAP compensation with the optimum number of eigenvectors per audio session, for both system and optimum clustering outputs.

the total eigenvalue mass as follows:

$$\arg\min_{k} \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{D} \lambda_j} \geq p \tag{4.15}$$

being D the dimension the within-class scatter matrix W.

Figure 4.31 shows the overall performance applying NAP and selecting k following Equation 4.15. This method seems to be more stable. However, the obtained DER is far from that optimum value shown in Table 4.6 (plotted as “SysOut tuned NAP” and “optOut tuned NAP”). This indicates that the selection of k based on a percentage of the eigenvalue mass, although better than using a constant k, is still far for providing the optimal values of k for each session. The best result obtained for the system output is 14.25% DER with $p = 20$, versus 15.15% DER of the baseline. In the case of the optimum output, we observe a similar behavior, providing a DER of 10.19% against the 10.46% of the baseline for $p = 20$.

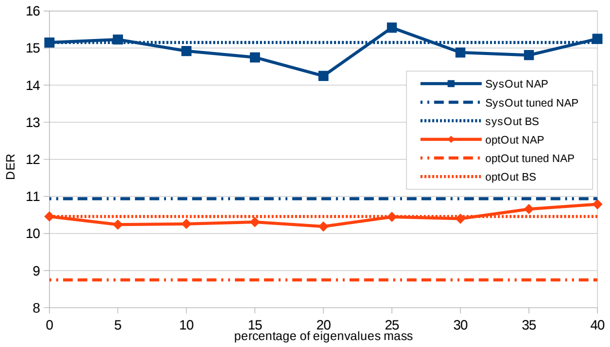


FIGURE 4.31: Diarization performance after applying NAP compensation, as a function of the percentage of eigenvalue mass used to decide the number of eigenvectors. Results are calculated for both system and optimum clusterings.

4.5 Conclusions

In this chapter, the main contributions of this thesis to the task of single-session speaker diarization have been described and assessed.

As result of some early work, a novel SAD approach based on binary keys was proposed and validated experimentally. Results suggest that there is still room for further improvement. Later, an alternative global clustering method able to implicitly determine the optimum number of speakers was adapted and introduced in our binary key speaker diarization system with the aim of replacing the faulty final clustering selection module.

Then, the core contributions of this thesis in single-session speaker diarization were exposed and evaluated. These contributions involve speeding up speaker diarization and improving overall performance. The KBM training process was optimized by replacing the KL2 distance by the cosine distance for measuring dissimilarity between Gaussian components. The modification has been shown to be beneficial in both terms of execution time and performance. The use of CVs instead of BKs for representing speech segments and cluster, and the use of the cosine and chi-squared similarities between CVs have resulted in additional performance gains. But the modification which contributes most to improving performance is the proposed final clustering selection technique. Up to now, the weak operation of the original clustering selection technique made the system rather unusable in real applications. Now, after the introduction of the proposed improvements, this issue is significantly mitigated.

With regard to the proposed session variability compensation approach, the obtained gains in performance are very subtle. Furthermore, estimating the within-cluster scatter matrix penalizes execution time severely since the CVs extracted from the development data must be re-computed for each new test session. Therefore, one must wonder if it is worthy to get minimum improvements in performance at the cost of a great increase in execution time. In principle, this basis departs from the goals of this thesis.

In general, after applying all the proposed improvements we have obtained a very fast yet reasonably accurate diarization system, more suitable than other state-of-the-art systems for processing large collections of audio files. In terms of execution time, our approach outperforms those reviewed in Section 2.3.3. It is also important to note that neither external data nor background resource computing are involved in any of the stages of the diarization (excluding session compensation).

Chapter 5

Cross-session binary key speaker diarization

This chapter describes the research done in this thesis in the topic of cross-session speaker diarization. As it has been already discussed in the state of the art chapter, cross-session speaker diarization is an extension to the classical problem of speaker diarization, in which recurrent speakers who participate across different audio sessions must be labeled using the same abstract identifiers in all sessions where they participate. In other words, each speaker present in a given collection must be assigned a unique and collection-wise speaker ID. This task is very useful in situations where a subset of speakers are common across a multimedia archive. This is the case, for example, of TV and radio programs, where there are common speakers such as presenters, journalists and actors. In those situations, it is highly desirable to assign the same IDs to those speakers across the entire collection.

In the literature, this task is mainly referred to as cross-show speaker diarization. The term "show" is used because this task has mainly been investigated in the context of TV and radio broadcast data. However, the task could also be applied to audiovisual data of different nature, such as phone calls, meeting recording, or any other audio kind with multiple and recurrent speakers involved. For this reason, we extend the term of cross-show speaker diarization to cross-session speaker diarization, as we find this generalization more appropriate.

In Chapter 2, the main approaches to cross-session speaker diarization were reviewed. Those approaches include the global approach by concatenation, the hybrid approach, and the incremental approach. The three approaches have their pros and cons. The global approach by concatenation is the simplest one and consists in concatenating all the sessions and performing speaker diarization as usual. The main problem of this approach is the vast memory and computational requirements when the collection becomes longer and longer. The hybrid approach partially solves this issue by performing single-session speaker diarization on each session and then grouping the resulting clusters in a collection-wise fashion. This approach may also present issues when the collection grows over time,

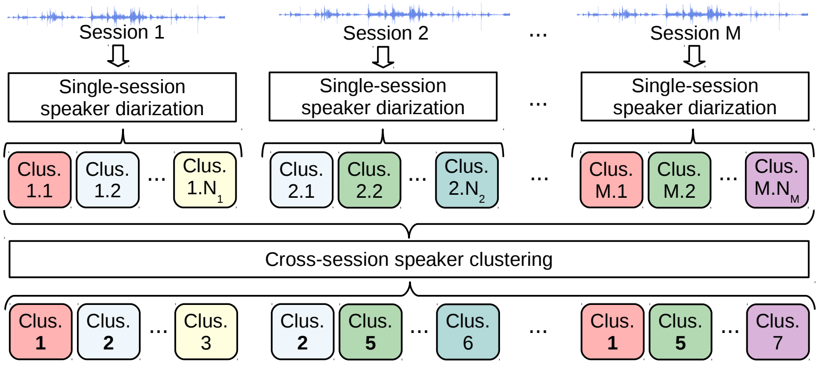


FIGURE 5.1: Hybrid cross-session speaker diarization architecture.

as the global clustering has to be repeated entirely when there are new incoming sessions. Finally, the incremental approach tries to face this problem by using the information of the current sessions in the collection to perform speaker diarization on the incoming sessions. The problem of this approach is that a database of speakers has to be maintained in order to decide whether there are new speakers in the incoming sessions or not. In practice, this is equivalent to an open-set speaker identification system.

One of the main goals of this thesis is to provide competitive execution times in order to process audio collections as fast as possible. In this regard, we have obtained a fast single-session speaker diarization system which is based on the binary key speaker modeling. This system could be directly applied to the task of cross-session speaker diarization by following the approach by concatenation, but it would suffer from computational issues when processing big collections. Instead, we propose to follow a hybrid scheme based on binary key speaker modeling, where first each session is processed separately through our single-session binary key system, and then the resulting speaker clusters, represented by cluster binary keys, are subject to a global clustering process which aims at finding the recurrent speakers. Finally, each detected unique speaker is assigned a global speaker ID. In addition, we explore how to compensate the inter-session variability previous to the global clustering stage.

The rest of the chapter is structured as follows: Section 5.1 describes the proposed binary key cross-session speaker diarization system. Section 5.2 evaluates our proposal. Section 5.3 compares the obtained results with other works in the state of the art. Finally, Section 5.4 provides some conclusions.

5.1 System description

As said above, the proposed method follows an hybrid approach where first each session is diarized separately using our binary key based speaker diarization system, and then all resulting clusters are clustered to conform the final cross-session clustering. Figure 5.1 illustrates a hybrid cross-session architecture. The clusters derived from single-session speaker diarization are assigned unique, session-wise abstract IDs. Next, the cross-session

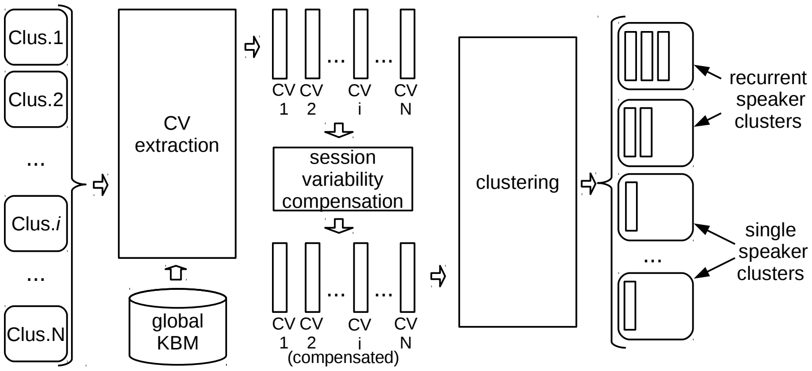


FIGURE 5.2: Proposed binary key cross-session speaker diarization system.

speaker clustering aims at grouping those clusters from different sessions which belong to the same speaker (cross-session speakers, represented with the same colors in Figure 5.1). Once the possible recurrent speakers have been identified and assigned a collection-wise ID, the speaker labels already obtained from the single-session speaker diarization processes are edited and replaced by the collection-wise speaker IDs.

Figure 5.2 depicts the architecture of our proposed binary key based cross-session speaker diarization system. We can separate the cross-session diarization scheme into three parts. First, each returned cluster from the individual diarizations must be converted into CVs. Second, inter-session variability compensation on the CVs is applied. Finally, the obtained compensated vectors are clustered to obtain clusters of speakers participating in more than one session. On one hand, CVs estimated on non-recurrent speakers will ideally constitute single clusters on their own. On the other hand, CVs from recurrent speakers will ideally be grouped together into recurrent speaker clusters.

It has been commented in previous paragraphs that the hybrid approach to cross-session speaker diarization can suffer from memory and computational issues if the data collection is big. Our approach faces these problems in two different ways:

- **Each session is first processed through the fast binary key speaker diarization system**. Although this system comprises an Agglomerative Hierarchical Clustering (AHC) stage, it has been shown in Chapter 4 that the single-session speaker diarization system presents very competitive execution times. However, AHC complexity is not linear, but cubic (or cuadratic in the best case, according to the implementation). It can be argued that this fact may result in too long processing times for big data collections. Nevertheless, session lengths are usually bounded from above, and it can be expected that the single-session diarization will be performed within a reasonable time.
- **The global clustering is performed over a set of CVs representing the output speaker clusters derived from the single-session process**. In this way, a whole cluster which can be of several minutes of data is compacted into a single, relatively low-dimensional vector. As a result, a collection of several hours

> of audio content can be compacted into a set of several hundred vectors. As it will be shown later, performing a clustering stage on this reduced amount of vectors is feasible and fast, and therefore the technique is applicable to collections of several hours.

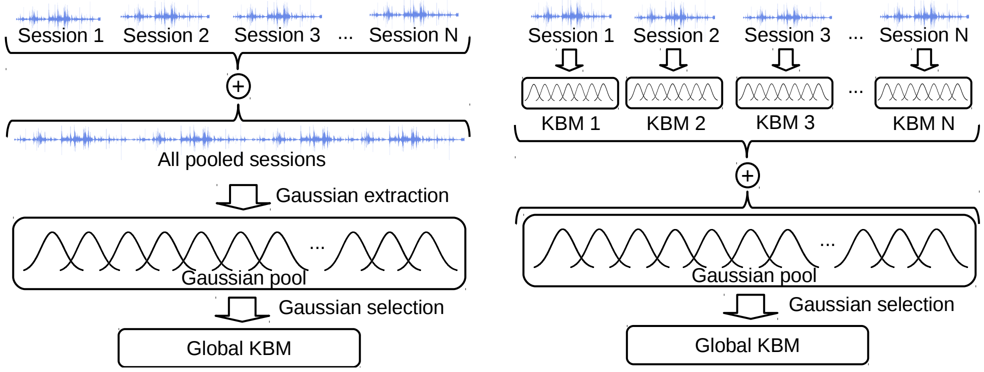
But there is still a problem that must be addressed. One could think that the BKs/CVs already obtained in the single-session diarization could be used in the global cross-session clustering. This situation would be highly desirable since all the necessary resources would have been created in the single-session step and could be reused directly in the global stage. Unfortunately, this is not possible because the BKs/CVs obtained from a given session were computed by using its session-dependent KBM. This fact makes BKs/CVs not comparable along different sessions.

Therefore, it is required to estimate cluster BKs which can be compared along the collection. This implies the need of a common, global KBM which could be used in the estimation of all cluster BKs for the entire collection. In this work we propose two different approaches to build this cross-session KBM. These approaches are described in Section 5.1.1. Next, two different method for the cross-session clustering are proposed in Section 5.1.2. Finally, Section 5.1.3 shows how to apply inter-session variability compensation on the cumulative vector space.

5.1.1 Cross-session KBM

In order to estimate CVs of the input clusters, a collection-wise KBM is required so that the obtained vectors can be compared in a cross-session fashion. For this purpose, two different methods for estimating the cross-session KBM are proposed. This methods are illustrated in Figure 5.3. The first one (Figure 5.3a), which we call "global KBM training" first pools all the input sessions together to get a virtual audio stream containing all sessions. Then, the cross-session KBM is obtained by just applying the KBM training algorithm as it is used for single-session speaker diarization: first, a pool of Gaussian components is obtained by using a sliding window with some window shift, and second, the Gaussian selection process is launched in order to obtain the target number of Gaussians which will make up the final KBM. The second method (Figure 5.3b), which we call "bootstrapped KBM training" take advantage of the single KBMs generated in the single-session diarization stage (which could be saved at the moment of single-diarization). The Gaussians of those KBMs are pooled together and then, the Gaussian selection stage is applied over that pool of Gaussians. We use the term "bootstrapped" because the initial Gaussian pool is taken from the individual single-show diarizations, thus there is no need to estimate new Gaussians on all sessions for the cross-session process.

Once the KBM is obtained by following one of the proposed methods, each input cluster can be easily converted into BKs/CVs as usual by using the computation method described in Section 3.1.2.



(A) Global KBM training.

(B) Global bootstrapped KBM training.

Figure 5.3: Schemes of the two proposed methods for estimating the cross-session KBM training.

5.1.2 Cross-session speaker clustering

The aim of the cross-session speaker clustering stage is to group the input CVs, obtained from session-dependent speaker clusters, into unique speaker clusters along the entire collection of sessions. Two different clustering methods are proposed for this task. The fist one is the classical Agglomerative Hierarchical Clustering (AHC). The second is the ILP clustering method proposed by [Rouvier and Meignier, 2012], already tested for single-session speaker diarization in this thesis (Section 4.1.2).

5.1.2.1 Agglomerative hierarchical clustering

As it has been already discussed in the state of the art, AHC has been widely used in the task of single-session speaker diarization as a speaker clustering technique. Commonly, the input stream is first divided into a set of N_{init} initial clusters, where N is greater than the actual number of speakers. From each cluster, a cluster model is obtained through direct training on the cluster data, or through model adaptation. Then, the data is redistributed into the set of clusters (for example, by maximum likelihood). Finally, the resulting clusters are evaluated and the closest cluster pair is merged, reducing the total number of clusters by 1. This process is applied iteratively until a stopping criterion is met, or until the number of clusters is 1. This kind of speaker clustering is used on the single-session binary key based speaker diarization system under study in this thesis.

However, this re-training/re-assignment scheme may not be appropriate within our proposed cross-session speaker diarization scheme because, once the input single-session speaker clusters have been converted into collection-wise CVs, the access to the original data is lost, and therefore CVs for new cross-session speaker clusters cannot be trained by using the BK/CV estimation method described in Section 3.1.2. In place of using cluster models for comparing cluster pairs, distances between them can be calculated as a function of the distances between the members of the two clusters. This clustering method is known as cluster linkage. Three common linkage criteria are the single linkage, the complete linkage, and the average linkage.

In the single linkage criterion, the distance between two clusters A and B is defined as

$$D_{single}(A, B) = \min\{D(a, b) | a \in A, b \in B\} \tag{5.1}$$

In other words, the distance between clusters A and B is equal to the distance of the closest pair of points in the two clusters. Oppositely, the complete linkage defines the distance between clusters A, and B as

$$D_{complete}(A, B) = \max\{D(a, b) | a \in A, b \in B\} \tag{5.2}$$

that is, the distances between A and B is equal to the distance between the farthest pair of points in the two clusters. Finally, the average linkage defines the distance between A and B as

$$D_{avg} = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} D(a, b) \tag{5.3}$$

In this case the distance is equal to the average distance between all pairs of points in the two clusters. These clustering methods enables us to ignore the speech segments within each input cluster, and then new formed cross-session speaker clusters can be effectively compared between them without the need of estimating cross-session cluster models.

As in single-session speaker diarization, the AHC process must be stopped at some point. In this case, a simple threshold can be used as stopping criterion.

5.1.2.2 ILP clustering

One of the main limitations of AHC is its greedy nature. Cluster merges are decided according to local optima, which may lead to clustering solutions that are not optimal in a global point of view. Recently, an alternative approach to the classic AHC for speaker clustering was presented in [Rouvier and Meignier, 2012]. That work proposes a global clustering scheme set out as an optimization problem of Integer Linear Programming (ILP), in which the aim is to minimize the number of final clusters and the dispersion within those clusters at the same time. In Chapter 4, this method was introduced within our binary key speaker diarization system in order as an attempt to replace the original faulty clustering selection algorithm.

As it has been explained in Section 4.1.2, this approach is intended to be applied after a first clustering stage which ideally produces an under-clustered set of highly pure clusters. Then, each obtained cluster is converted into a single vector (e.g. i-vector or binary key) which represent it. Finally, the ILP clustering is performed over those vectors in order to build the final set of clusters.

In a hybrid cross-session speaker diarization scheme, clusters obtained from individual speaker diarization over the individual audio files have to be clustered together according to the recurrent speakers in the collection of sessions. If each input cluster is represented by a single vector, the proposed ILP clustering can be straightforwardly applied. Therefore, this approach seem to adequately fit the cross-session diarization framework. In fact, the ILP clustering has been successfully applied in [Dupuy et al., 2012], and further studied in [Dupuy et al., 2014a,b].

In Section 4.1.2, the ILP clustering method for single-session speaker diarization was described. The goal is to group the N input clusters (already converted into single points) into K clusters. Each resulting cluster consists of a center plus a set of points associated to that center (the set can be empty). The role of each point in the problem must be determined in order to minimize the number of clusters and the within-cluster dispersion. To this end, some constraints and binary variables have to be defined (consult Section 4.1.2). In this case, we adopt the ILP proposal described in [Dupuy et al., 2014a], with overall distance filtering, which is much more efficient than the original formulation, as the number of variables and constraints is significantly reduced (and therefore the search space too), leading to computational savings. The motivation here is that very dissimilar point pairs are not likely going to be associated, and then all those possible solutions of the problem can be safely discarded. This approach requires the previous computation of the matrix of all pair-wise distances for all points, but this will allow to remove Equation 4.5 from the ILP formulation. After applying the filtering, the ILP clustering can be reformulated as:

let $C \in \{1, ..., N\}$, $K_{j \in C} = \{k / d(k,j) < \delta\}$

Minimize

$$\sum_{k \in C} x_{k,k} + \frac{1}{D} \sum_{k \in K_j} \sum_{j \in C} d(k,j) x_{k,j} \tag{5.4}$$

Subject to

$$x_{k,j} \in \{0,1\} \qquad k \in K_j, j \in C \tag{5.5}$$

$$\sum_{k \in K_j} x_{k,j} = 1 \qquad j \in C \tag{5.6}$$

$$x_{k,j} - x_{k,k} < 0 \qquad k \in K_j, j \in C \tag{5.7}$$

As distances are now implicitly taken into account, Equation 4.5 may be removed. K_j is the set of possible values of k which satisfy the distance restriction between center k and point j (refer to Section 4.1.2 for further details).

The ILP approach can be easily adapted to our binary key cross-session speaker diarization framework. First, the output clusters generated at each single-session speaker diarization can be converted into BKs or CVs by means of a global KBM (previously estimated by following one of the methods described in Section 5.1.1). Next, the obtained BKs/CVs are used as input points for the ILP clustering.

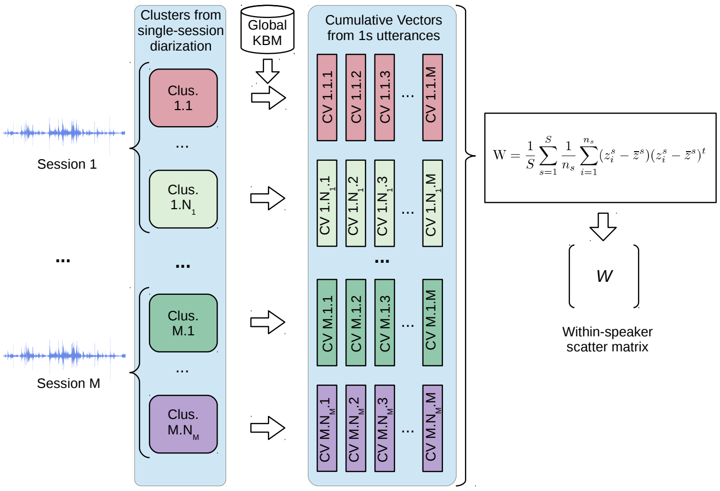


Figure 5.4: Process for obtaining the within-speaker covariance matrix W from cumulative vectors representing 1s utterances from the clusters returned by the single-session diarizations. CVs are calculated using the global KBM previously estimated.

5.1.3 Inter-session variability compensation

It has already been discussed in Section 4.3.4 how inter-session variability has a negative impact on performance of speaker recognition systems. For this reason, session variability compensation techniques have been proposed in the last years. Some of these techniques such as Joint Factor Analysis (JFA) and the most recent i-vector paradigm have achieved impressive improvements in performance.

In the task of single-session speaker diarization, there is no actual inter-session variability, since the process is performed on a single session. However, the acoustic conditions can vary within a session. This variability is referred to as intra-session variability. In Section 4.3.4, we proposed to use the nuisance attribute projection (NAP) compensation method over the cumulative vector space in order to deal with intra-session variability.

Turning to the task of cross-session speaker diarization, the concept of inter-session variability becomes meaningful again, since the recurrent speakers appearing in different sessions could have been recorded under different acoustic conditions. Therefore, it seems reasonable to argue that compensating session variability may result beneficial in order to remove the influence of the changing acoustic conditions in the process of finding recurrent speakers among a collection of audio sessions.

In this regard, we propose the use of NAP compensation applied on the cumulative vector space. We tested this method for compensating intra-session variability for the single-session speaker diarization task (Section 4.4.2). The evaluation results showed partial success. We only say "partial" because, although performance gains were obtained, the requirement of development data to estimate the required within-class scatter matrix

	REPERE test set	REPERE dev set
#Clusters	212	196
#Speakers	145	140
#Recurrent speakers	36	32

Table 5.1: Number of speaker clusters, actual number of speakers, and number of cross-session speakers, of the REPERE test and development sets.

W is out of the scope of the main goals of this thesis, due to its supervised nature and its important penalty on execution time.

In order to overcome this limitation, we propose a method to estimate W in an unsupervised way. This method is illustrated in Figure 5.4. As it has been pointed out before, estimating W requires labeled data with speaker information. We propose to estimate W on the test data itself by using the obtained segmentations on the single-session speaker diarization processes as speaker labels. Given a session, all speech assigned to speaker i is concatenated and divided into segments of 1 second. Each of those segments is treated as a speaker utterance in the calculation of W. The process is repeated over all speaker clusters of all individual diarizations until obtaining speaker labels for the whole collection. Before applying Equation 4.12, CVs have to be estimated for all 1 second segments using the global KBM. Note that estimating W for the whole collection does not impose too much computation load, since it is computed only once thanks to the only global KBM (recall that W had to be computed over and over again for each single-session speaker diarization using their own KBMs).

5.2 Evaluation

In this section, the proposed approach to cross-session speaker diarization based on binary keys is assessed. The main elements to be evaluated are listed below:

- Cross-session KBM: global KBM and bootstrapped KBM.
- Cross-session speaker clustering: AHC and ILP.
- Linkage method for AHC: single, complete and average.
- Inter-session variability compensation.

As in the evaluation of the single-session speaker diarization system, all the proposals are first evaluated on the REPERE phase 1 test set. Once the best performing system and configuration is found, such system will be evaluated on a different unseen dataset consisting on the REPERE phase 1 development set. This will be done in order to check the stability of system configuration across different datasets.

Table 5.1 shows information about the speakers of the REPERE test and development datasets. "#Clusters" refers to the number of clusters counted on the reference speaker segmentation files independently. "#Speakers" refers to the actual number of

	REPERE test	REPERE dev
#Clusters (#clusters in ref.)	139 (212)	133 (196)
single-session DER (%)	15.15	18.17
base cross-session DER (%)	26.88	35.40

Table 5.2: Information about the single-session clustering solutions used as input clusters for the cross-session diarization system, on the REPERE test and development datasets: number of clusters, DER of single-session diarization, and baseline cross-session DER (i.e. cross-session DER if each input cluster is considered to represent a unique speaker).

speakers in the whole collection. Finally "#Recurrent speakers" is the number of the speakers (from those specified in the second row) who participate in more than one show. Ideally, the cross-session diarization process should take #Clusters clusters as input clusters, and the result should be a set of #Speakers clusters, where #Recurrent speakers - #Speakers clusters should correspond to the non-recurrent speakers, whilst the remaining #Recurrent speakers clusters should correspond to the recurrent speaker clusters.

As stated before, our proposal follows a hybrid scheme where each session is first diarized separately, and then the resulting clusters are used as input data for the cross-session speaker diarization. Therefore, we take the output clusters obtained by the best single-session system found in Chapter 4. Such single-session speaker diarization system exhibited performances of 15.15% and 18.17% DER on the REPERE test and development sets, respectively. The main points of that system are summarized next:

- KBM size of 320 components.
- Cosine similarity as similarity metric between cumulative vectors.
- Uniform cluster initialization.
- Final clustering selection method based on within-cluster sum of squares and elbow criterion.

Table 5.2 collects some data about the set of clusters generated on the single-session speaker diarization: number of clusters (and, in brackets, the actual number of clusters according to the ground-truth reference labels), their corresponding single-session DER, and the baseline cross-session DER as if each input cluster were considered as a different speaker ("base cross-session DER" row) on the REPERE test and development sets. As it is shown, the number of clusters returned by the single-session speaker diarization process (139 and 133 on the test and development sets, respectively) is smaller than the actual number of clusters in the reference files (212 and 196 on the test and development sets, respectively). This means that the clustering selection stage of the single-session speaker diarization system tends to return clustering solutions with fewer clusters than the actual number of clusters in the reference. Figures shown in the third row are useful in order to set a performance floor, corresponding to the performance when no cross-session speaker clustering is performed at all.

5.2.1 Cross-session speaker clustering

This subsection evaluates the two proposed methods for cross-session speaker clustering, namely the AHC method and the ILP clustering method.

5.2.1.1 Cross-session AHC

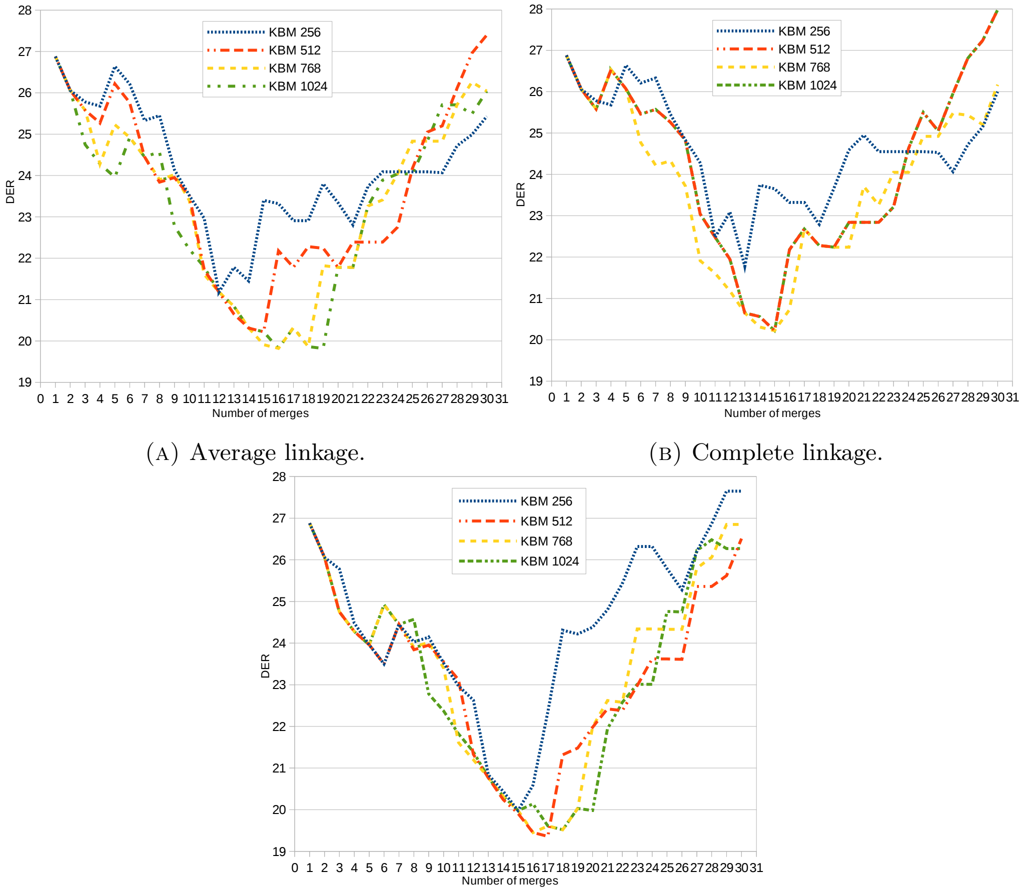
One of the first points to be checked is if the speakers in all sessions are effectively modeled by CVs generated using the global KBM. In other words, the global KBM should produce speaker CVs comparable along the whole collection of sessions. If so, CVs estimated on clusters of the same speaker should exhibit higher similarity (or equivalently, shorter distance or dissimilarity) values than pairs of CVs estimated on clusters from different speakers. Therefore, we expect that, within an AHC process, the first merges will group all CVs from recurrent speakers. This would result in a decrease of DER while the recurrent speaker will be being grouped together. After that, DER will start to increase in subsequent iterations as long as incorrect merges will be being performed. In addition, we are interested in exploring the three proposed linkage criteria, namely average, complete, and single linkage. And finally, the last part under evaluation is the estimation of the cross-session KBM, for what two methods have been proposed, namely the global KBM training method, and the bootstrapped KBM training method.

In order to asses all the mentioned in the previous paragraph, the first experiment aims at evaluating performance with regard to the number of iterations of the AHC, using the three proposed clustering criteria (average, complete, and single linkages), for several KBM sizes. Figure 5.5 shows these results for the global KBM training method, while Figure 5.6 shows the same for the bootstrapped KBM training. Regardless on the linkage criterion and the KBM training used, DER decreases after several initial AHC iterations. Ideally, this first section of the DER curve should be monotonically decreasing until all clusters of recurrent speakers have been clustered together. However we can see that there are some increasing peaks in this area. This means that some incorrect merges are being produced. A second observation common to all clustering criteria and KBM training methods concerns the KBM size. In all situations, bigger KBMs seem to yield more accurate cluster CVs, what results in better clustering performance. KBM sizes of 768 and 1024 show similar performance, and KBM sizes greater than 1024 do not result in performance gains. Third, if we focus on the linkage criteria, it seems that the single linkage provides slightly better performance for both KBM training methods. Finally, Table 5.3 shows DER according to the used linkage criterion and the used KBM training method for a KBM size of 768, when the AHC process is manually stopped at the optimum iteration (i.e. the one providing the best DER value). From this result, we deduce that the best performing combination involves the use of the global KBM training and the single linkage criterion, providing a DER value of 19.45%.

This first experiment has been useful in order to determine the most suitable linkage criterion and KBM training method. However, DER has been plotted with regard to the number of iterations of the AHC. This does not enable an automatic method for deciding when the AHC must be stopped. In this sense, it is more useful to study DER with regard to a threshold on the maximum distance allowed to perform cluster merges. If at the

	Average link	Complete link	Single link
Global KBM	19.82	20.22	**19.45**
Boot. KBM	20.32	20.73	19.86

Table 5.3: DER (%) obtained at the optimum AHC iteration (selected manually) with regard to the KBM training method and the linkage criterion used, for a KBM size of 768 Gaussian components.

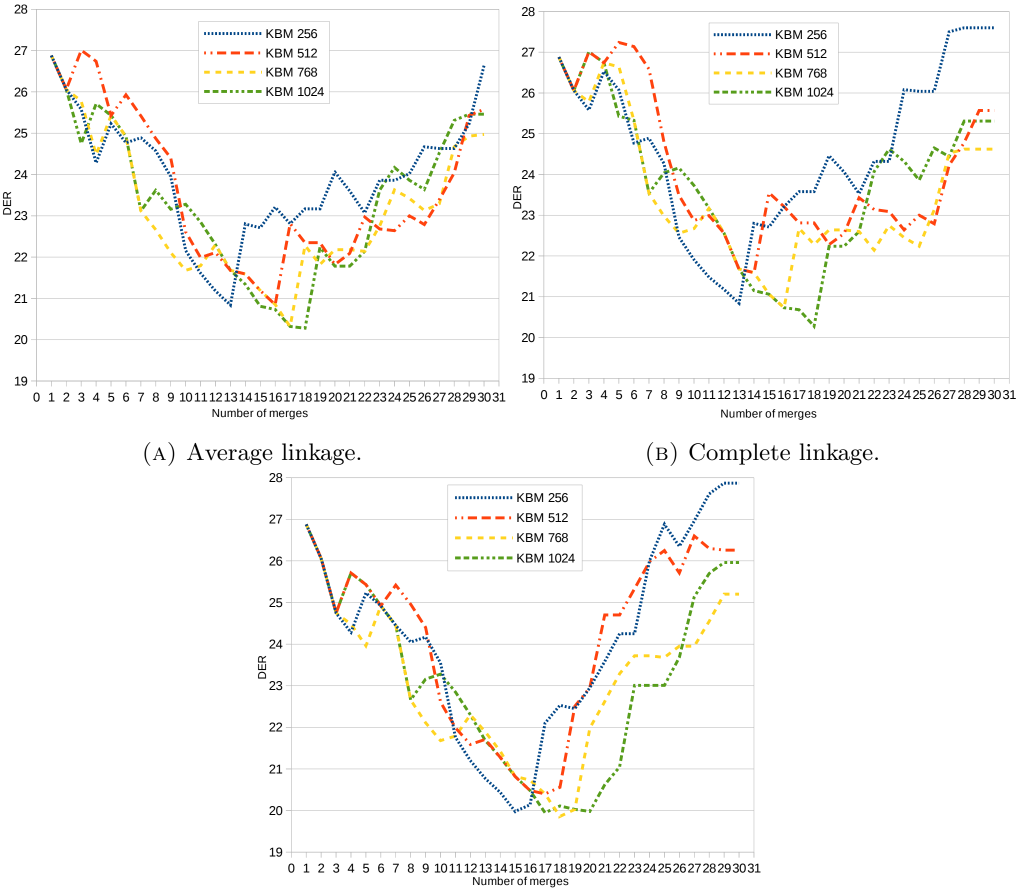


(a) Average linkage.

(b) Complete linkage.

(c) Single linkage.

Figure 5.5: Evaluation of the **global KBM** training method: DER with regard to the number of AHC iterations, for the three proposed linkage criteria (average, complete, and single), and for different KBM sizes.



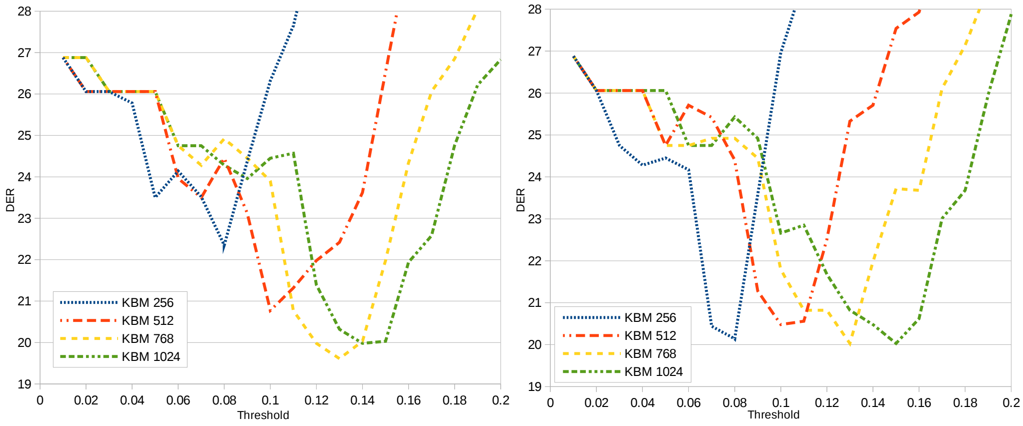
(A) Average linkage.

(B) Complete linkage.

(C) Single linkage.

FIGURE 5.6: Evaluation of the **bootstrapped KBM** training method: DER with regard to the number of AHC iterations, for the three proposed linkage criteria (average, complete, and single), and for different KBM sizes.

current iteration all distances between current clusters are bigger than the threshold, then the process is be stopped. Figure 5.7 shows DER with regard to the threshold used to stop the AHC, for different KBM sizes of 256, 512, 768, and 1024. The results are shown for the case of the global KBM training method (Figure 5.7a) and for the case of the bootstrapped KBM training method (Figure 5.7b). The best result is obtained using the global KBM training. However, the bootstrapped KBM shows better performance stability across different KBM sizes. A second observation is the dependency of the threshold value on the KBM size. The smaller the KBM is, the shorter the optimum threshold is. This fact was already observed in the analysis of BKs and CVs conducted in Chapter 4: within-speaker similarities tend to be smaller (and therefore distances longer). This explains the need of incrementing the threshold value when bigger KBMs are being used. After these results, we establish the best performing AHC system as the one using a KBM of 768 components trained using the global training method, the single



(A) Global KBM.

(B) Bootstrapped KBM.

FIGURE 5.7: Evaluation of **cross-session AHC** with regard to the threshold δ, for different KBM sizes and the two KBM training proposals.

		AHC		**ILP**	
	DER floor (%)	δ_{opt}	DER (%)	δ_{opt}	DER (%)
Global KBM	26.88	0.13	**19.61**	0.13	**19.45**
Boot. KBM	26.88	0.13	20.03	0.13	19.86

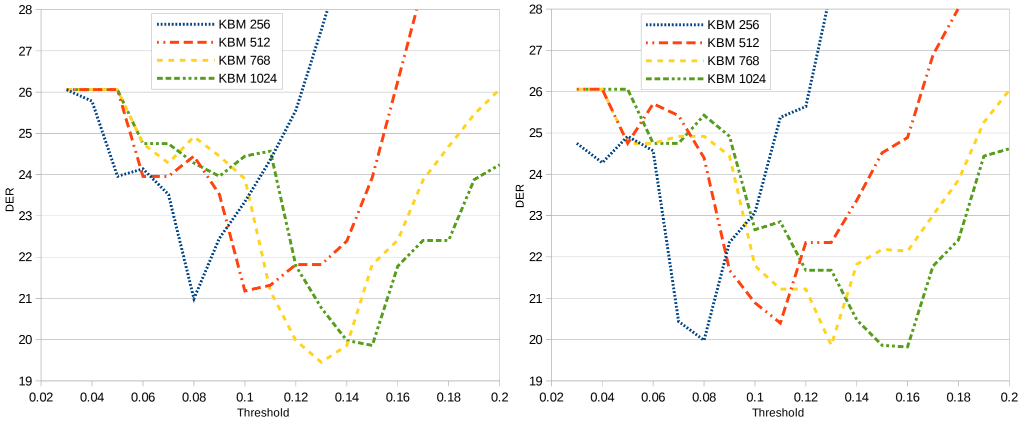
TABLE 5.4: DER (%) obtained with the AHC and ILP clustering approaches using the optimum threshold for KBMs of 768 components trained with the global and bootstrapping methods.

linkage criterion, and a threshold $\delta = 0.13$ on the maximum distance between clusters as stopping criterion.

5.2.1.2 Cross-session ILP clustering

After evaluating the cross-session AHC clustering method, it is the turn of assessing the cross-session ILP clustering. Figure 5.8 shows performance of the ILP clustering with regard to the threshold δ on the maximum distance to allow points to be associated to a center. Performance is calculated for different KBM sizes and for the two proposed KBM training methods. If we compare the obtained results with the ones obtained with AHC (Figure 5.7), it can be seen how performances are very similar for all combinations of KBM training and KBM sizes. Even the optimum thresholds are compatible for both AHC and ILP techniques.

Table 5.4 compares performance of the AHC and ILP methods for a KBM size of 768. $\delta_{opt} = 0.13$ for the 4 possible combinations of KBM training and speaker clustering. Once again, the global KBM training presents slightly better performance than the bootstrapped one, providing DER figures of 19.61% and 19.45% with the AHC and ILP clustering methods, respectively. Therefore, AHC and ILP seem to be practically equivalent in terms of performance. Even the threshold values seem compatible between



(A) Global KBM.

(B) Bootstrapped KBM.

FIGURE 5.8: Evaluation of **cross-session ILP clustering** with regard to the threshold δ, for different KBM sizes and the two KBM training proposals.

the two approaches. The obtained cross-session performance is just around 4% absolute higher than its single-session counterpart. This difference in performance is common in different works over the same dataset [Dupuy et al., 2012; Rouvier et al., 2013; Galibert and Kahn, 2013].

5.2.2 Inter-session variability compensation

The cross-session speaker clustering methods evaluated in the previous subsection has been shown to be quite effective for the detection and clustering of those input clusters from recurrent speakers. The aim of this subsection is to check if compensating inter-session variability could be helpful for the task of cross-session speaker clustering, as it has been hypothesized. We take the best performing configuration for cross-session speaker clustering found above (KBM size of 768, $\delta = 0.13$) and we perform NAP compensation in order to obtain compensated cluster CVs. After estimating the within-speaker scatter matrix W on the set of input clusters as explained in Section 5.1.3, the projection is applied to the set of input cluster CVs by following Equation 4.14.

Figure 5.9 plots cross-session speaker diarization DER when applying NAP session variability compensation, with regard to the NAP order k ($k = 0$ means no session variability compensation). Recall that k is the number of top eigenvectors used to construct the NAP projection. DER is shown for the 4 possible combinations of speaker clustering methods (AHC and ILP) and KBM training methods (global and bootstrapped). Let us first focus on the system using the global KBM training. In this case, NAP results beneficial for values of k between 1 and 9 for both clustering approaches, leading to slight DER improvements up to around 1% absolute DER. However, if we put the focus on the system using the bootstrapped KBM training, NAP produces the opposite effect and DER becomes higher. As for the optimum NAP order for the system with global KBM training, $k = 6$ for AHC, while $k = 3$ for ILP.

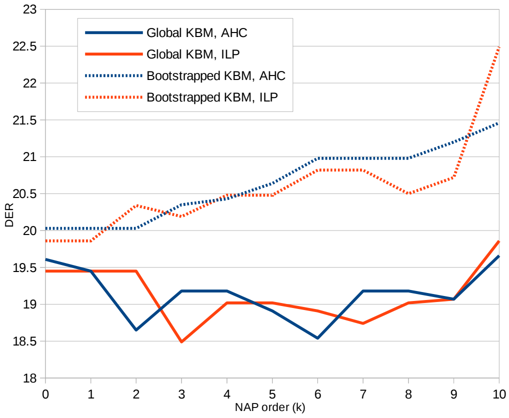


FIGURE 5.9: DER of session-compensated cross-session speaker diarization with regard to the NAP order k (number of eigenvectors used to build the projection matrix), for the AHC and ILP clusterings, and KBMs of 768 components trained using the two different KBM estimation methods.

Data set	**Single DER**	**DER floor**	**DER AHC**		**DER ILP**	
	-	-	w/o NAP	NAP	w/o NAP	NAP
REPERE test	15.15	26.88	19.61	**18.54**	19.45	**18.49**
REPERE dev	18.17	35.40	25.93	**24.34**	24.71	**24.34**

TABLE 5.5: Cross-session diarization performance measured in DER with KBM size of 768, $\delta = 0.13$, with and without NAP session variability compensation, obtained on the REPERE test and development sets. The input single-session and floor cross-session diarization performances are also included as reference.

5.2.3 Testing the system on a different dataset

The previous experiments have shown the effectiveness of the proposed binary key cross-session architecture. Two different KBM training procedures have been proposed and evaluated, being the called global KBM training the most accurate. In addition, two different clustering methods have been proposed and assessed, namely AHC and ILP clustering. As for AHC, three different linkage criteria have been evaluated, namely average, complete, and single linkages. The single linkage criterion has resulted to be the best performing in our scheme. AHC and ILP have resulted to provide very similar performance, being practically equivalent for the task. Finally, a first attempt to compensate inter-session variability compensation on the cumulative vector space has been performed, contributing with a slight improvement in performance.

All these results have been obtained on the REPERE phase 1 test dataset. Although the proposed approach does not require development data, all the proposals and improvements have been tested on the same dataset. As we already did for single-session speaker diarization, it would be very interesting to evaluate the best performing system found on a new dataset in order to check if performance is stable over new unseen data. To this end, we run the best performing systems using AHC and ILP speaker

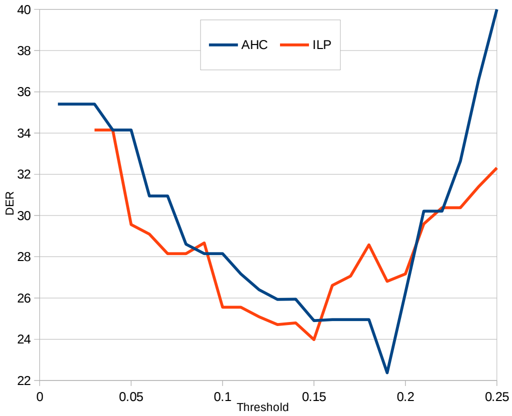


FIGURE 5.10: Cross-session performance on the REPERE development set measured in DER for the AHC and ILP clustering methods, with regard to the threshold δ.

clustering, with NAP session compensation. The parameters were set to the optimum values obtained for the REPERE test dataset: KBM size of 768, $\delta = 0.13$, and $k = 6$ and $k = 3$ for the AHC and ILP clustering methods, respectively. Table 5.5 collects the obtained results on the REPERE test and development set together. The columns "Single DER" and "DER floor" are included to provide a context in terms of DER: the first is the average, time-weighted DER of the input single-session diarizations, while the second correspond to the DER calculated on a hypothetical cross-session clustering in which each cluster is considered as a unique speaker. For the REPERE development set, the obtained cross-session diarization without session compensation is 25.93% and 24.71% for the AHC and ILP clustering methods, respectively. Therefore, ILP seems to behave slightly better on this database. However, after applying NAP, both clustering approaches obtain the same performance figure of 24.34% DER. The increment of DER with regard to the single-session baseline performance (third column of Table 5.5) is of 6.17% absolute DER, against the 3.39% absolute DER obtained on the REPERE test set. Therefore, the increment is approximately twice as high.

In order to check if the higher increment of error is due to the use of a not optimal threshold for the new dataset, Figure 5.10 plots DER on the REPERE development set with regard to the threshold δ. Several remarks can be extracted here. First, unlike on the REPERE test set, δ are different for each clustering method: $\delta = 0.15$ for ILP, while $\delta = 0.19$ for AHC. Second, also in contrast to the test set, performances delivered by ILP and AHC differ more, being the AHC clustering more effective here, providing an absolute improvement of 1.61% DER (22.37% DER for AHC against 23.98% DER for ILP). Furthermore, as we were suspecting, the optimum δ values estimated on the REPERE test set are not optimal on the REPERE development set. In fact, performance improves around 1.97% absolute DER, even if no session compensation is applied on the last test. Finally, it is interesting to remark how the DER curve of the AHC system keeps monotonically decreasing until the optimum DER is reach. This means that all the performed cluster merges, from the beginning until the optimum DER is reached, are correct.

Stage	**xRT AHC**	**xRT ILP**
Single diarization	0.0354	
KBM estimation	0.00007	
CV estimation	0.0329	
NAP matrix estimation	0.0039	
Cross-session clustering	0.0000646	0.0001349
Total Cross-session	**0.0368**	**0.0369**
Single + Cross	**0.0722**	**0.0723**

Table 5.6: Execution time of cross-session speaker diarization measured in real-time factor (xRT), broken down into the different stages of cross-session diarization. KBM size of cross-session speaker diarization is 768.

5.2.4 Execution time

Finally, Table 5.6 shows execution time figures of the cross-session diarization system using the two proposed clustering methods and the best configuration found above (KBM size of 768, $\delta = 0.13$), broken down into the main stages involved. We can see how the global KBM training is very fast, presenting xRT values of -5 order of magnitude (10^{-5}).

The next stage involves the computation of the cumulative vectors. This is, by far, the longest stage of the process, due to the extensive computation of likelihoods for all the feature vectors with regard to all Gaussian components in the KBM. This stage takes even more time than the prior single-session speaker diarization stage which produces the input clusters, because the use of a bigger KBM (768 components for cross-session diarization against 320 components for single-session diarization) penalizes execution time (as a reference, xRT of single-session diarization with a KBM of 320 component is 0.0354, while single-session diarization with a KBM size of 768 presents xRT of 0.0555). Preparation of CVs for cross-session with KBM size of 768 has xRT of 0.0488, which is consistent with the execution times of single-session diarization with the same KBM size (note that the AHC stage and the final clustering selection are not performed here, but just the computation of CVs, what explains the smaller xRT obtained).

The third step concerns the estimation of the NAP matrix, which supposes a xRT of 0.0067. Unlike NAP for single-session speaker diarization as proposed in Section 4.3.4 the within-class scatter matrix W is computed only once on the input clusters coming from the prior single-session speaker diarization stage. In the case of cross-session speaker diarization, this compensation method is not only feasible, but perfectly applicable in terms of execution time.

Finally, execution time of cross-speaker clustering for the AHC and ILP methods are shown. AHC seem to be more efficient than the ILP method with xRT orders of magnitude of -5 against -4, respectively. Anyways, both methods are very fast for our purposes, at least with the number of input clusters used.

To sum up, the total xRT for the cross-session speaker diarization task can be computed as the sum of all xRT of each step. Cross-session execution time using the AHC method is 0.0368 xRT, while execution time using the ILP method is 0.0369. Both methods present virtually the same execution times, although AHC has shown to present one

			AHC		ILP	
Factor	**#Clusters**	**Collection duration (h)**	**Execution time (s)**	**xRT**	**Execution time (s)**	**xRT**
x1	139	2.81	0.13	1.27×10^{-5}	1.03	1.02×10^{-4}
x2	278	5.63	0.06	3.02×10^{-6}	1.13	5.56×10^{-5}
x3	417	8.44	0.11	3.72×10^{-6}	1.06	3.48×10^{-5}
x4	556	11.26	0.19	4.59×10^{-6}	1.90	4.69×10^{-5}
x8	1112	22.51	0.75	9.29×10^{-6}	12.98	1.60×10^{-4}
x16	2224	45.03	3.35	2.07×10^{-5}	-	-
x32	4448	90.05	13.88	4.28×10^{-5}	-	-
x64	8896	180.11	55.46	8.55×10^{-5}	-	-
x128	17792	360.21	223.23	1.72×10^{-4}	-	-

Table 5.7: Study of the impact of the collection size on the cross-session speaker clustering.

order of magnitude less. In spite of that, clustering times are negligible with regard to the heaviest part of computing the CVs. The total time took by the two diarization process (single- and cross-session speaker diarization) is 0.0722 xRT and 0.0723 for AHC and ILP clustering methods, respectively. Although the cross-session stage takes slightly more execution time than the single-session speaker diarization stage, we still find this execution times very reasonable to perform the combination of single- plus cross-session diarization processes.

Speaker clustering scalability

All the experiments described above were performed on a dataset of around 3 hours. Execution times obtained in such amount of data are satisfactory. The success of the used clustering techniques, in terms of computation, relies on the fact that several hours of data can be represented by only a few hundreds of vectors. For those amounts of data points, the polynomial complexity of the algorithms result in practical execution times. However, if we increase the number of vectors of the problem, there could be a point when the resulting execution times become unpractical. This point will be reached when the order of magnitude of the cross-speaker clustering is no longer negligible with regard to the order of magnitude of the most expensive stage of CV estimation.

In order to assess the scalability of the proposed cross-session speaker clustering techniques, we have simulated a scenario dealing with bigger data amounts. The original dataset of around three ours could be replicated several times in order to increase the number of initial clusters and the duration of the whole collection. Then, the speaker clustering can be run in isolation in order to study the scalability and the impact of data size on execution time. Table 5.7 shows the factor used to enlarge the source dataset, the number of input clusters, the duration of the resulting collection in hours, and then execution time and xRT for the AHC and ILP clustering approaches. Figure 5.11 also plots the xRT values of xRT columns of Table 5.7.

First, let us analyze the AHC speaker clustering. When the collection is incremented up to a factor of 8, the order of magnitude keeps at -6. When using a factor of 64, the order of magnitude is -5, which is still far from the order of magnitude of the CV

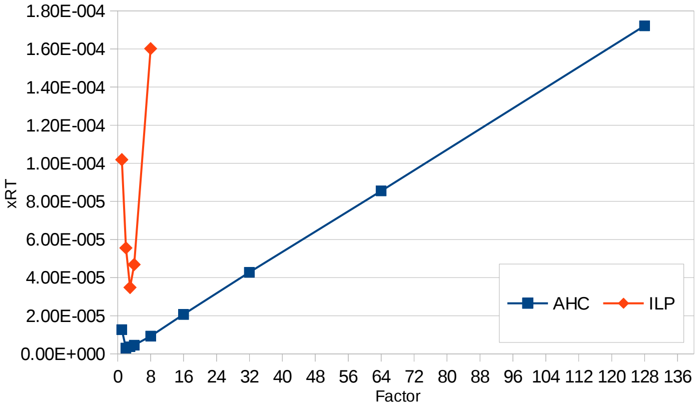


FIGURE 5.11: Evolution of xRT (Y axis) of cross-session speaker clustering when multiplying the size of the input dataset by Factor (X axis).

preparation stage (-2). Finally, using a factor of 64, the order of magnitude is -4. Using this factor, the number of input clusters is of 17792, and the collection length is 360.21 hours. These results indicate that there is still room for increasing the number of input clusters to several tens of thousands until the order of magnitude of -2 is reached. The absolute time needed to run the clustering on such amount of input points is 223.23s, which is a very reasonable time with regard to the actual amount of audio data.

Second, let us focus on the ILP approach. As it was already shown, the ILP method is more expensive than the AHC, and shows one more order of magnitude in xRT (-5 up to factor of 4, and -4 for factor of 8). We could not extend the experiment to bigger collections because of a lack of memory. The ILP problem has to be defined in terms of an objective function and a set of constraints. The number of constraints can grow significantly with the number of input data, and the available RAM memory used for the experiments (8 GB) does not allow to define such number of constraints. However, we believe that the ILP approach is still suitable in the context of big audio archives processed by powerful servers with more RAM memory available.

In summary, this experiment shows the feasibility of the proposed cross-session speaker clustering schemes, at least for processing collections of several hundreds of hours while maintaining very competitive execution times.

Further accelerating cross-session speaker diarization

Although the achieved execution times for cross-session speaker diarization are very competitive, it has been found that the main bottleneck of the method is the estimation of the CVs on the input clusters. This step is mandatory since the CVs generated on the single-session processes cannot be used in a global context, as CVs estimated on different KBMs are not comparable in collection-wise manner. Therefore, it is necessary to estimate new collection-wise CVs to be used for cross-session speaker clustering.

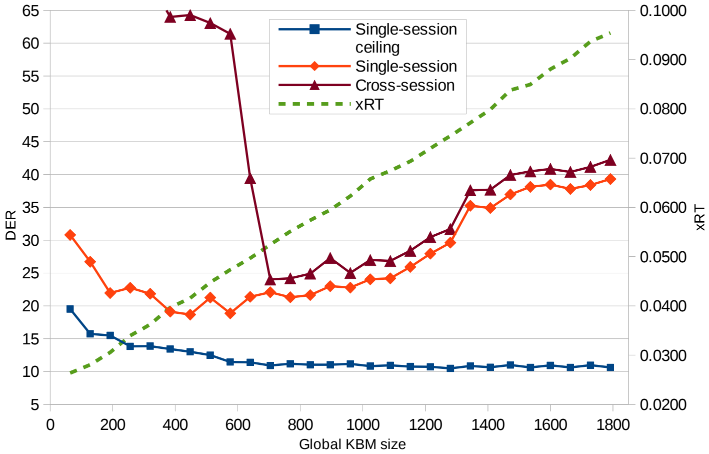


Figure 5.12: Results of single- and cross-session speaker diarization using the same global KBM in both phases: DER of single-session performance ceiling, DER of single-session speaker diarization, and DER of cross-session speaker diarization. Execution time measured in real-time factor (xRT) of the overall process is also provided (secondary Y axis).

A possible alternative which would allow reusing the CVs obtained on the single-session stage is to directly use a common, global KBM estimated on the whole collection on each individual single-session diarization. In this way, the obtained local CVs are generated by the same KBM and become globally comparable.

In order to explore this alternative, first the global KBM can be computed on all the concatenated sessions in the collection (as it is done for the global KBM for cross-session). Then, this KBM can be used on the individual single-session diarization. This may result in computational savings, as there is no need to compute KBMs for each session. However, this method will presumably be less accurate since each session will have fewer representation in the global KBM than in *ad hoc*, single-session KBM. Intuitively, bigger KBM sizes will be necessary to provide a strong modeling of speakers in a single-session basis. In addition, using bigger KBM will have an impact on single-session execution time, but this increment will likely be compensated by the gains achieved in the cross-session stage.

Figure 5.12 shows the obtained results of performing single-session speaker diarization using a unique global KBM for all sessions. First, in order to assess the feasibility of the speaker modeling by the global KBM for single-session speaker diarization, the performance ceiling is shown ("Single-session ceiling"). Recall that this ceiling is the performance obtained by selecting the best clustering solution of each session manually. It can be seen that DER converges around 10%. However, comparing to the performance ceiling of the regular single-session speaker diarization system, bigger KBMs are required in order to get similar performance. According to this result, the global KBM for single-session shows potential at the cost of a slightly bigger KBM (and subsequently an increase in execution time).

Second, single-session performance is plotted ("Single-session"). The DER curve shape shows similar patterns as those found in the regular single-session case (see Figure 4.26). However, performance is, for all KBM sizes, significantly worse than the regular case. The best DER achieved is 18.87% with a KBM of 576 components, while the best DER of the best performing configuration of the regular single-session system was 15.15%. This result indicates than the selection of the final clustering solution is quite sensitive to the use of the global KBM. If the clustering selection could be improved in order to provide clustering solutions close to the performance ceiling, then the proposed scheme could make sense.

Third, the clusters obtained in the single-session stage are passed to the cross-session speaker diarization system. Here, the CVs generated on the single-session step can be directly used, so that we remove the global CV computation stage, and the speaker clustering can be directly performed. The obtained cross-session performance is plotted by the "Cross-session" series. Small KBMs do not seem to provide good global CVs, and performance is extremely low. At KBM size of 768, cross-session DER curve behaves very similarly to the single-session DER curve, with an approximately constant increment of around 2% in DER. Once again, this result confirms that the proposed approach is meaningful and consistent with the single-session clustering provided as input. However, the single-session best clustering returning should be improved in order to get performance close to the regular proposed cross-session scheme. The best cross-session performance achieved is 24.02% DER, which is significantly higher than performance obtained by the regular cross-session system using the same clustering method (19.61% DER).

Let us not forget that the main motivation of this last approach is to share the resources generated in the single-session diarization in order to obtain gains in execution time. The dashed line in Figure 5.12 ("xRT", secondary Y axis) shows the real time factor with regard to the KBM size. xRT shows linear behavior. For the best performing configuration (KBM of 768 components) xRT is 0.0551. This supposes an absolute improvement on xRT of 0.0171 xRT with regard to the best performance achieved on the regular single- plus cross-session system. However, performance suffers significantly and it should be evaluated if the gains in execution time are worthy at the cost of important accuracy degradation. As pointed out before, this method could be suitable if the previous single-session speaker diarization with global KBM, and particularly the selection of the final clustering solution, could provide clustering outputs with accuracy close to the performance ceiling.

5.3 Comparison of performance with the state-of-the-art

Up to now, all the proposals to single- and cross-session binary key speaker diarization have been evaluated, obtaining gains in both performance and speed compared to the baseline system. In this section we review single-session and cross-session speaker diarization results reported on several works on the REPERE database, and we compare them with the results obtained in this thesis. There are just a few works reporting results with this database, which are shown in Table 5.8. These works/systems are listed below:

- "Official REPERE team A, B and C" refer to the official results achieved by the 3 participants of the REPERE challenge. These results are reported in [Galibert and Kahn, 2013].
- "LIUM@Interspeech2013" refers to the LIUM[1] system results reported in [Rouvier et al., 2013].
- "LIUM@Odissey2014" refers to the LIUM system results reported in [Dupuy et al., 2014a].

Let us start with the participant systems at the official evaluation. The three participants on the official REPERE evaluation were:

- SODA consortium, formed by LIUM (Laboratoire d'Informatique de l'Université du Maine), and IDIAP Research Institute.
- QCOMPERE consortium, formed by LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur), INRIA research center, LIG (Laboratoire d'Informatique de Grenoble), YACAST, Vocapia Research, GREYC (Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen), and KIT (Karlsruhe Institute of Technology).
- PERCOL consortium, formed by LIF (Laboratoire d'Informatique Fondamentale de Marseille), UAPV (Université d'Avignon et des Pays de Vaucluse), LIFL (Laboratoire d'Informatique Fondamentale de Lille), and France Télécom.

The speaker diarization systems presented by the three consortia are briefly review next. The SODA speaker diarization system is that developed by LIUM [Rouvier et al., 2013]. This system first perform a GLR (generalized likelihood ratio) plus BIC speaker segmentation and an AHC speaker clustering using BIC as a distance measure between clusters, modeled by full covariance Gaussian distributions. Then, a Viterbi re-segmentation using GMMs as speaker models is performed in order to refine segment boundaries. Next, a speech/non-speech segmentation is obtained using Viterbi decoding in order to remove music and jingle regions. At this point, a number of highly pure clusters greater than the actual number of speakers is obtained. A final clustering stage based on i-vectors and ILP method is performed. An i-vector is extracted from each cluster using 19 MFCCs parameters completed with energy, their first and second order derivatives, and a 1024 GMM-UBM. The resulting i-vectors are then normalized in order to cope with the intra-session variability. Finally, the i-vectors are subject to an ILP clustering. With regard to cross-session speaker diarization, an additional ILP clustering is performed on i-vectors extracted from each output cluster from the previous single-session speaker diarization stages.

The system presented by the QCOMPERE consortium comprises two different diarization systems. The first one is the LIMSI speaker diarization system [Barras et al., 2006]. This system relies in two steps: agglomerative clustering based on the BIC criterion to provide pure clusters, followed by a second clustering stage using cross-likelihood

[1] Laboratoire d'Informatique de l'Université du Maine

System	Single-session DER (%)	Cross-session DER (%)
Official REPERE team A	13.70	33.09
Official REPERE team B	13.35	16.05
Official REPERE team C	**11.10**	**14.20**
Official REPERE team C (only ILP)	**14.37**	**17.69**
LIUM@Interspeech2013 AHC/CLR	17.19	23.95
LIUM@Interspeech2013 ILP/i-vector	15.46	19.59
LIUM@Odyssey2014 AHC/CLR	16.22	-
LIUM@Odyssey2014 ILP/i-vector	14.60	-
Binary key	**15.15**	**18.49**

Table 5.8: Single-session and cross-session speaker diarization performance results of several systems in the literature, on the REPERE Phase 1 test set.

ratio (CLR) as distance between the clusters. For cross-session speaker diarization, a local clustering stage is followed by a CLR clustering across all the sessions. The second system is the one developed by KIT. First, speech/non-speech detection is performed based on a GMM-HMM segmenter. Then a two-pass speaker segmentation is performed, the first one based on BIC, and the second one performed as a Viterbi re-segmentation stage. The speaker models are trained on the obtained clustering results. Then. feature warping is applied for compensating channel effects, and GMMs with 64 mixtures are used for modeling the speakers. Finally, a BIC clustering and a Viterbi re-segmentation are applied.

The system presented by the PERCOL consortium is that developed by France Telecom [Charlet et al., 2013]. A first step performs an agglomerative clustering of speech segments based on Bayesian Information Criterion (where each cluster is modeled by a single Gaussian with a full covariance matrix). When each cluster contains enough data to model the voice more precisely, the clusters are modeled with Gaussians mixture, and the agglomerative clustering is performed with a distance between clusters based on a cross-likelihood criterion. At each iteration of the clustering based on cross-likelihood, a Viterbi decoding is also performed to re-segment the speech data into speaker turns, given the new clusters. In addition this system pays special attention to overlapping speech. Overlapping speech segments are first detected and discarded from the clustering process, and at the end, the remaining segments with overlapping speech are assigned to the 2 nearest speakers in terms of temporal distance between speech segments.

In [Galibert and Kahn, 2013] the official REPERE results are reported. However, the participant partners were renamed as teams A, B, and C in order to preserve their anonymity. However, we have known that the winner team (C team) corresponds to the SODA consortium. As it can be seen in Table 5.8, participant teams in the official REPERE evaluation achieved very good results in the task of single-session speaker diarization, with DER figures ranging from 13.70% to 11.10%. With regard to cross-session speaker diarization, with the exception of team A with 33.09% DER, results are very close to the single-session ones (16.05% and 14.20% DER).

If we compare the results obtained in this thesis (“Binary key” row of Table 5.8) with the ones obtained by the winner system, we see that our single-session binary key system DER is a 4.05% absolute above, and that our cross-session system DER is a 4.39% absolute above. However, we have known that the winner system performed a

Stage	xRT
BIC segmentation + clustering	0.04
Viterbi re-segmentation	0.03
ILP/i-vector clustering (single-session)	0.05
Overall single-session	**0.12**
ILP/i-vector clustering (cross-session)	0.02
Overall single- plus cross-session	**0.14**
Binary key single-session	0.035
Binary key cross-session	0.036
Binary key single- plus cross-session	0.072

Table 5.9: Execution times (xRT) of the LIUM speaker diarization system, decomposed into the different stages, on a 2.4 GHz Intel Core i5 processor, on the REPERE corpus. The three last rows show execution times obtained by the binary key speaker diarization systems for comparison.

combination of speaker diarization, overlapping speech detection and speaker identification, which provided the final results (11.1% DER and 14.2% DER for single- and cross-session speaker diarization, respectively). As our binary key system does not perform any of those supporting tasks, the comparison of performance between both systems is not totally accurate. Performance obtained by the winner system by only applying speaker diarization[2] is shown in Table 5.8 ("Official REPERE team C (only ILP)" row). In single-session speaker diarization, our system error is just a 0.78% absolute above, while in cross-session speaker diarization, our system error is just a 0.8% absolute above.

The rest of the results shown in Table 5.8 are extracted from other publications by LIUM. Essentially, the system described in those works is the same presented in the REPERE evaluation, but without incorporating overlapping speech handling and speaker identification. In addition to the ILP speaker clustering, results for a AHC/-CLR speaker clustering are also included. The ILP method seem to outperform the AHC/CLR method, providing performance improvements of around 2% absolute DER on single-session diarization, and around 4% absolute DER on cross-session diarization. Performances achieved in this thesis are very close to those reported in these papers. However, let us not forget that our results have been obtained using ground-truth speech/non-speech labels, and that this dataset was used for the development of our systems. Therefore, in a real scenario, the difference in performance between the two systems would probably be slightly higher in favor of the LIUM system. Even so, the main conclusion extracted is that our system performance is very close to other system performances of the state of the art, even though our system is significantly simpler that those reviewed here.

Although not formally published in the literature, one can find some execution time figures on the LIUM speaker diarization web site[3]. Table 5.9 collects execution times measured in terms of real-time factor (xRT), decomposed into the different stages involved.

[2] Special thanks to Sylvain Meignier from LIUM for kindly providing these internal results.

[3] `http://www-lium.univ-lemans.fr/diarization/doku.php/overview`, accessed on 28/05/2015

In order to compare execution times between our binary key system and the LIUM system, we have removed the execution times of feature extraction and speech/non-speech detection stages from the LIUM system.

As it can be seen, the most time consuming stages are the BIC segmentation and clustering (0.4xRT), and the ILP/i-vector clustering (0.05xRT). Our single-session speaker diarization system is even faster (0.035) than any of those two stages. One of the factors that makes the binary key system faster is the absence of a dedicated speaker segmentation stage: our system follows an ICSI-like approach where data re-assignment is performed after each iteration of the agglomerative clustering. Since these data reassignments are based on simple similarity measures between binary keys or cumulative vectors, the process is very efficient. With regard to the ILP clustering execution time showed in Table 5.9, this figure must be interpreted cautiously, because later published work on ILP clustering [Dupuy et al., 2014a] reports great computation savings by reformulating the ILP clustering problem as an exploration of a connected graph, obtaining real-time factors of 0.0011xRT.

As for cross-session speaker diarization, the LIUM system presents a slightly better real-time factor than the binary key based cross-session system (0.02xRT against 0.036xRT). This is caused by the additional stage required to re-calculate the cumulative vectors for the input clusters (recall that the cumulative vectors generated at each single-session speaker diarization cannot be used in the global clustering). However, the cross-session AHC clustering of the binary key system presents a real-time factor of 0.0000646xRT, and it has been shown to scale to several thousands of input vectors (see Table 5.7). Our cross-session speaker clustering approach seems to outperform (in terms of execution time) the optimized version of the cross-session ILP/i-vector speaker clustering.

5.4 Conclusions

In this chapter, our investigations on the task of cross-session speaker diarization using the binary key speaker modeling have been presented.

After a brief introduction to the task, a new system for cross-session speaker diarization has been proposed. This system is also based on the binary key speaker modeling approach, which has been shown to be an efficient method for processing large data collections. Our proposal follows a hybrid architecture in which a set of speaker clusters derived from a first single-session speaker diarization stage are converted to single binary keys. This is a key point since each cluster (which can be of several minutes) is compacted into only one vector. This allows to represent hours of audio data into just several hundreds of vectors. Then, a clustering stage is performed in order to group those vectors belonging to recurrent speakers who participate in more than one session within the collection. Furthermore, inter-session variability compensation on the cumulative vector space has been performed through NAP compensation method with unsupervised estimation of the within-class covariance matrix.

Estimating binary keys or cumulative vectors requires the use of a KBM. Two different methods for estimating a cross-session KBM have been proposed: the global and the bootstrapped ones. While the global method trains the KBM on all the data in the collection from scratch, the bootstrapped method takes the individual KBM obtained on each single-session speaker diarization in order to reduce the number of initial Gaussian components. Experiments showed that KBMs trained through the global method provides slightly better performance than KBMs obtained by the bootstrapped method.

As for speaker clustering, first a classic AHC clustering was tested with three different linkage criteria: single, complete, and average. In our system, the single linkage is the best performing one against the complete and average linkages. Second, the ILP global clustering method was assessed. Both AHC and ILP methods provide practically the same performance, but the AHC is significantly more efficient in terms of execution time and memory usage.

Regarding inter-session variability, our unsupervised method for estimating the within-class scatter matrix has resulted beneficial, providing slight performance improvements at the cost of just a subtle increase on execution time.

The best performing systems uses a global KBM of 768 components, trained following the global method. For both AHC and ILP clustering methods, the threshold on the maximum distance used to determine the final number of clusters is $\delta = 0.13$. However, the optimum NAP order k (number of eigenvectors used to estimate the NAP projection) is different for each clustering technique, being $k = 6$ for AHC and $k = 3$ for ILP. Performance obtained with AHC was $DER = 18.54\%$, while performance with ILP was $DER = 18.49\%$. These DER values are just around 3% absolute above DER of the input single-session clustering solution (DER=15.15%).

After identifying the best configuration found on the REPERE test set, the REPERE development set was processed by the system with the same optimum settings in order to evaluate performance stability across different data. In this case, the obtained DER is 24.34% for both clustering methods. This result is around 6% absolute higher than the input single-session performance of 18.17% DER. It has been observed than estimated parameters on the REPERE test dataset are not optimum for the REPERE development set, being $\delta = 0.19$ for AHC, and $\delta = 0.15$ for ILP, the optimum values for this dataset. Using those optimum parameters, performance with AHC is 22.37% DER and performance with ILP is 23.98% DER.

The proposed method has been shown to be efficient on the REPERE datasets. These collections have around 3 hours duration each one. The performed study on the scalability of the cross-session speaker clustering shows that the approach is also suitable for collections of several hundreds of hours. Although the AHC clustering has $O(n^2)$ complexity, it has been shown that the technique is feasible for a set of several hundreds of input vectors which may represent several hours of data.

The obtained execution time of 0.0722 xRT is already quite good for the complete task of single- plus cross-session speaker diarization. This xRT figure is broken down into 0.0354 xRT from single-session speaker diarization and 0.0368 xRT from cross-session speaker diarization. Both tasks show very similar execution times. However, a big part of the time of cross-session diarization is used to calculate the global resources needed, particularly the input CVs. This stage is performed because the cluster CVs generated on the individual single-session processes cannot be used in a collection-wise way since those CVs are estimated using local KBMs. An interesting and promising method to avoid the re-computation of CVs consists in using a global, collection-wise KBM in all single-session diarizations. In this way, the CVs estimated on the local speaker diarization processes can be directly used in the cross-session speaker clustering stage. Therefore, the global CV estimation process can be removed from the cross-session stage, with what we save the majority of time used in the cross-session diarization. We have shown that this approach allows to reduce xRT by 0.0171 absolute with regard to the best performing cross-session system. However, this gain in execution time has the cost of a decrease in performance of around 5% absolute DER. The overall increase of error seems to come from the single-session stage, as it has been shown that the use of the global KBM require higher KBM sizes in order to get similar performance as using a local KBM, and that the final clustering selection does not perform as adequate with those bigger KBMs as for smaller KBM sizes. Anyways, this approach may be still useful if especial requirements on execution time are needed and a trade-off between execution time and performance can be achieved.

Finally, a comparison of the obtained results with those of the official REPERE evaluation, as well as with other works in the literature, indicates that performance of the proposed single- and cross-session systems based on binary keys are quite close to the state of the art, being the proposed systems considerably simpler and faster. We can conclude that the proposed systems may be more suitable for processing large amounts of data under special temporal requirements than other best-performing but slower ones.

Chapter 6

Conclusions

In this chapter the research carried out and the improvements obtained in the context of this thesis are reviewed. Finally, some possible lines for future work are discussed.

6.1 Overall review

This thesis is about the topic of fast cross-session speaker diarization for processing large collections of data with recurrent speakers. This challenge is faced in two different aspects. First, a fast yet accurate single-session speaker diarization system based on the binary key speaker modeling has been obtained. Second, a new binary key cross-session speaker diarization system has been proposed, developed and evaluated. The cross-session system takes the individual outputs from the single-session diarization of each session within the collection and performs a global speaker clustering which finds the recurrent speakers and assigns them a unique collection-wise ID. The system has been designed in order to process large collections in a reasonable time period.

In order to speed up single-session speaker diarization, the speaker diarization approach based on binary key speaker modeling described in [Anguera and Bonastre, 2011] was taken as a starting point. That paper reported execution times of around 0.1 xRT at the cost of a decrease in diarization performance. Our baseline system, based on this approach, was implemented and evaluated, obtaining consistent performances to the ones reported in the original work. The obtained results provided us the first clues about certain key points to be improved.

From those identified weak points, some modifications were proposed, always in order to improve both efficiency and accuracy. As for accuracy improvement, the use of cumulative vectors as speaker models was included together with some meaningful similarity measures which enable the effective comparison of pairs of cumulative vectors. Furthermore, a new criterion for the selection of the final clustering (and consequently, of the final number of speakers) was proposed in order to replace the original faulty criterion. As for execution time reduction, the Gaussian component selection procedure of the KBM training was lightened by replacing the KL2 distance by the cosine distance. After the experimental evaluation of the new system including all the improvements, a 16.8% relative improvement was achieved in terms of diarization performance (15.15% DER of the new system, versus 18.22% DER of the baseline system), being the new system around 7 times faster than the baseline (0.0345 xRT of the new system against 0.252 xRT of the baseline system). In addition, some attempts to intra-session variability compensation through the Nuisance Attribute Projection (NAP) on the cumulative vector space were performed, obtaining slight performance improvements, but at the cost of a considerable increase of execution time. Finally, and apart from speaker diarization, a new Speech Activity Detection system based on binary keys was proposed and successfully evaluated, obtaining similar or even better performance than a HMM-based state-of-the-art SAD system.

A new cross-session speaker diarization system based on binary keys was proposed. The basic principles of this system are borrowed from the single-session diarization system, but had to be adapted to the new task. Two different approaches for training a cross-session KBM, as well as two different cross-session speaker clustering approaches were proposed. Furthermore, unsupervised inter-session variability compensation was addressed through the NAP compensation technique. Experimental results show just slightly higher error rates (around 3.5% absolute DER) than the single-session clustering solutions taken as starting point. As for execution time, the cross-session stage presents xRT values similar to the ones of single-session speaker diarization system, being 0.0722 xRT the total execution time of the complete single- plus cross-session task. It was observed that the majority of time used by the cross-session system is for estimating global cumulative vectors for the input clusters. To face this issue, an alternative method which allows sharing the cumulative vectors from the single-session stage into the cross-session stage was explored, opening the doors to a very interesting line.

Performances achieved by the proposed systems are not very far from those obtained in the official REPERE evaluation and in other works in the literature. Those approaches rely on advanced speaker modeling techniques such as i-vectors or JFA, which relies on vast amounts of training and development data, required to train UBMs, total variability matrices, and other resources. In addition, participating systems on the official evaluation performed combinations of different technologies such as overlapping speech detection and speaker identification. Our system's performance is a bit worse than the official results, but surely exhibiting better execution times than the participating systems.

Some of the contributions of this thesis have been submitted to and/or published in international conferences and journals. These papers are listed below:

Conference papers

- Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2015c). Novel clustering selection criterion for fast binary key speaker diarization. In *Proc. INTERSPEECH*
- Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2015b). Improved binary key speaker diarization system. In *Proc. European Signal Processing Conference (EUSIPCO)*

- Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2014a). Global speaker clustering towards optimal stopping criterion in binary key speaker diarization. In *Proc. IberSPEECH*, pages 59–68

- Delgado, H. and Serrano, J. (2014). Albayzin 2014 Evaluation: TES-UAB Audio Segmentation System. In *Proc. IberSPEECH*

- Delgado, H., Fredouille, C., and Serrano, J. (2014b). Towards a complete binary key system for the speaker diarization task. In *Proc. INTERSPEECH*, pages 572–576

Journal papers

- Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2015a). Fast single- and cross-show speaker diarization using binary key speaker modeling. *Submitted*

- Castan, D., Tavarez, D., Lopez-Otero, P., Franco-Pedroso, J., Delgado, H., Navas, E., Docio-Fernández, L., Ramos, D., Serrano, J., Ortega, A., and Lleida, E. (2015). Albayzín-2014 evaluation: audio segmentation in broadcast news domain. *Submitted*

- Delgado, H., Matamala, A., and Serrano, J. (2015d). Speaker diarization and speech recognition in the semi-automatization of audio description: an exploratory study on future possibilities? *Submitted*

6.2 Future work lines

There are a number of possible further improvements to the binary key speaker diarization systems presented in this thesis.

KBM training

With regard to the KBM estimation parameters, we think that performance will probably benefit from using adaptive KBM parameters (KBM size, window length and overlap) according to the audio duration. For example, very short audio session could not require as many Gaussians as a longer audio session could.

A second aspect to take into account is the possible information redundancy across binary keys from different speakers. In this regard, the less informative Gaussians (for example Gaussian components of the KBM that are selected in all the binary keys, regardless of the represented speaker) could be avoided by the use of masks over the binary keys.

The use of a global KBM for single-session speaker diarization proposed in Section 5.2.4 opens a series of possible paths to be explored: for example, how to train a meaningful global KBM on the target collection, or even how to pre-train a KBM from a population of known speakers.

Agglomerative clustering in binary key single-session speaker diarization

With regard to the data assignment stage, the use of fixed-length segments of 1s duration does not allow to obtain very precise segment boundaries and speaker change points. We believe that using a Viterbi-like decoding in the binary key domain would undoubtedly result in better boundary precision, and consequently, in gains in accuracy. But this alternative decoding should be designed having speed rates always present.

Further speeding up speaker diarization

After all changes applied to the baseline speaker diarization system, execution time was considerable reduced. However, there are still some processes that could be further sped up. Currently, the most expensive part of the system is the intensive calculation of likelihoods for all the input feature vectors with all the Gaussians in the KBM. Following a similar argument as that discussed in the acceleration of the KBM training (which led to replace the KL2 distance with a lighter similarity measure), alternative and lighter similarity measures could be used in place of the Gaussian probability density function. For instance, each feature vector could be compared to the Gaussian mean vector only and omit the use of the covariance matrix.

Binary key cross-session speaker diarization

The proposed cross-session diarization system is adequate for processing big data collections. However, the system does not take into account the possible inclusion of new session within the collection, thus the whole cross-session process should be periodically performed on the resulting increasing collection in order to keep it updated. This is one of the limitations of the hybrid approach to cross-session speaker diarization. To face this problem, other architectures, such as the incremental approach, or newly proposed ones could be explored.

Other challenges still open

Outside the context of binary key speaker diarization, there are still a number of open challenges which affect current speaker diarization approaches in general. One of these challenges is the meaningful detection and handling of overlapping speech. Current methods are able to detect the presence of overlapping speech, but fail at detecting how many voices are active and at modeling the actual speakers involved. Once the overlapping regions are detected, those are excluded from the diarization stage and it is at the end when the possible speaker present in such regions are assigned the same labels as their neighbor speakers. A meaningful speaker modeling in overlapping situations is still to be found.

And finally, estimating the actual number of speakers and cluster is still one of the most important sources of errors. As far as our binary key diarization system concerns, the selection of the output clustering solution has been greatly improved, but it has been shown that there is a performance ceiling which is still far to be reached. A more accurate selection algorithm able to get closer to the performance ceiling would systematically lead to great improvements in diarization performance.

Bibliography

Ajmera, J., Lathoud, G., and McCowan, I. (2004). Clustering and segmenting speakers and their locations in meetings. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I–605–8 vol.1.

Ajmera, J. and Wooters, C. (2003). A robust speaker clustering algorithm. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 411–416.

Anguera, X., Aguilo, M., Wooters, C., Nadeu, C., and Hernando, J. (2006). Hybrid speech/non-speech detector applied to speaker diarization of meetings. In *Proc. IEEE Odyssey: The Speaker and Language Recognition Workshop*, pages 1–6.

Anguera, X. and Bonastre, J.-F. (2010). A novel speaker binary key derived from anchor models. In *Proc. INTERSPEECH*, pages 2118–2121.

Anguera, X. and Bonastre, J.-F. (2011). Fast speaker diarization based on binary keys. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4428–4431.

Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., and Vinyals, O. (2012a). Speaker diarization: A review of recent research. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(2):356–370.

Anguera, X., Movellan, E., and Ferrarons, M. (2012b). Emotions recognition using binary fingerprints. In *IberSPEECH*.

Anguera, X., Wooters, C., and Hernando, J. (2007). Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2011–2022.

Aronowitz, H. (2007). Segmental modeling for audio segmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–393–IV–396.

Aronowitz, H. (2010). Unsupervised compensation of intra-session intra-speaker variability for speaker diarization. In *Proc. ODYSSEY: The Speaker and Language Recognition Workshop.*

Attias, H. (2000). A variational Bayesian framework for graphical models. In *NIPS*, volume 12.

Barras, C., Zhu, X., Meignier, S., and Gauvain, J. (2006). Multistage speaker diarization of broadcast news. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1505–1512.

Boakye, K., Trueba-Hornero, B., Vinyals, O., and Friedland, G. (2008). Overlapped speech detection for improved speaker diarization in multiparty meetings. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. (ICASSP)*, pages 4353–4356.

Bonastre, J., Bousquet, P., Matrouf, D., and Anguera, X. (2011). Discriminant binary data representation for speaker recognition. In *Proc IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5284–5287.

Bonastre, J.-F., Delacourt, P., Fredouille, C., Merlin, T., and Wellekens, C. (2000). A speaker tracking system based on speaker turn detection for NIST evaluation. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II1177–II1180 vol.2.

Bousquet, P. and Bonastre, J.-F. (2012). Typicality extraction in a speaker binary keys model. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1713–1716.

Campbell, W., Sturim, D., and Reynolds, D. (2006). Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., and Vair, C. (2008). Stream-based speaker segmentation using speaker factors and eigenvoices. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4133–4136.

Castan, D., Tavarez, D., Lopez-Otero, P., Franco-Pedroso, J., Delgado, H., Navas, E., Docio-Fernández, L., Ramos, D., Serrano, J., Ortega, A., and Lleida, E. (2015). Albayzín-2014 evaluation: audio segmentation in broadcast news domain. *Submitted.*

Castán, D., Vaquero, C., Ortega, A., González, D. M., Villalba, J. A., and Lleida, E. (2011). Hierarchical audio segmentation with HMM and factor analysis in broadcast news domain. In *Proc. INTERSPEECH*, pages 421–424.

Charlet, D., Barras, C., and Lienard, J.-S. (2013). Impact of overlapping speech detection on speaker diarization for broadcast news and debates. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7707–7711.

Chen, S. S. and Gopalakrishnan, P. S. (1998). Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion. pages 127–132.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Delacourt, P. and Wellekens, C. (2000). Distbic: A speaker-based segmentation for audio data indexing. *Speech Communication*, 32(1–2):111 – 126. Accessing Information in Spoken Audio.

Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2014a). Global speaker clustering towards optimal stopping criterion in binary key speaker diarization. In *Proc. IberSPEECH*, pages 59–68.

Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2015a). Fast single- and cross-show speaker diarization using binary key speaker modeling. *Submitted.*

Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2015b). Improved binary key speaker diarization system. In *Proc. European Signal Processing Conference (EUSIPCO).*

Delgado, H., Anguera, X., Fredouille, C., and Serrano, J. (2015c). Novel clustering selection criterion for fast binary key speaker diarization. In *Proc. INTERSPEECH.*

Delgado, H., Fredouille, C., and Serrano, J. (2014b). Towards a complete binary key system for the speaker diarization task. In *Proc. INTERSPEECH*, pages 572–576.

Delgado, H., Matamala, A., and Serrano, J. (2015d). Speaker diarization and speech recognition in the semi-automatization of audio description: an exploratory study on future possibilities? *Submitted.*

Delgado, H. and Serrano, J. (2014). Albayzin 2014 Evaluation: TES-UAB Audio Segmentation System. In *Proc. IberSPEECH.*

Dupuy, G., Meignier, S., Deléglise, P., and Estève, Y. a. (2014a). Recent improvements on ILP-based clustering for broadcast news speaker diarization. In *Proc. Odyssey: The Speaker and Language Recognition Workshop*, Joensuu (Finland).

Dupuy, G., Meignier, S., and Estève, Y. (2014b). Is incremental cross-show speaker diarization efficient for processing large volumes of data? In *Proc. INTERSPEECH*, pages 587–591.

Dupuy, G., Rouvier, M., Meignier, S., and Estève, Y. (2012). I-vectors and ILP clustering adapted to cross-show speaker diarization. In *Proc. INTERSPEECH.*

Ellis, D. P. W. and Liu, J. C. (2004). Speaker turn segmentation based on between-channel differences. In *Proc. NIST Meeting Recognition Workshop at ICASSP 2004.*

Evans, N., Fredouille, C., and Bonastre, J.-F. (2009). Speaker diarization using unsupervised discriminant analysis of inter-channel delay features. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4061–4064.

Ferras, M. and Boudard, H. (2012). Speaker diarization and linking of large corpora. In *Proc. SLT*, pages 280–285.

Fredouille, C., Bozonnet, S., and Evans, N. W. D. (2009). The LIA-EURECOM RT'09 Speaker Diarization System. In *RT 2009, NIST Rich Transcription Workshop, May 28-29, 2009, Melbourne, USA*, Melbourne, UNITED STATES.

Fredouille, C. and Evans, N. (2008). The LIA RT'07 speaker diarization system. In Stiefelhagen, R., Bowers, R., and Fiscus, J., editors, *Multimodal Technologies for Perception of Humans*, volume 4625 of *Lecture Notes in Computer Science*, pages 520–532. Springer Berlin Heidelberg.

Fredouille, C. and Evans, N. W. D. (2007). The influence of speech activity detection and overlap on speaker diarization for meeting room recordings. In *Proc. INTERSPEECH*, pages 2953–2956.

Fredouille, C. and Senay, G. (2006). Technical improvements of the E-HMM based speaker diarization system for meeting records. In Renals, S., Bengio, S., and Fiscus, J. G., editors, *MLMI*, volume 4299 of *Lecture Notes in Computer Science*, pages 359–370. Springer.

Friedland, G., Hung, H., and Yeo, C. (2009a). Multi-modal speaker diarization of real-world meetings using compressed-domain video features. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4069–4072.

Friedland, G., Vinyals, O., Huang, Y., and Muller, C. (2009b). Prosodic and other long-term features for speaker diarization. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):985–993.

Galibert, O. (2013). Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. In *Proc. INTERSPEECH*, pages 1131–1134. ISCA.

Galibert, O. and Kahn, J. (2013). The First Official REPERE Evaluation. In *Proc. SLAM@ INTERSPEECH.*

Garcia-Romero, D. and Espy-Wilson, C. Y. (2011). Analysis of i-vector length normalization in speaker recognition systems. In *Proc. INTERSPEECH*, pages 249–252.

Gauvain, J.-L., Lamel, L., Adda, G., and Jardino, M. (1999). The LIMSI 1998 Hub-4E Transcription System. In *Proc. DARPA Broadcast News Workshop*, pages 99–104.

Ghaemmaghami, H., Dean, D., Vogt, R., and Sridharan, S. (2011). Extending the task of diarization to speaker attribution. In *Proc. INTERSPEECH*, pages 1049–1052.

Gonina, E., Friedland, G., Cook, H., and Keutzer, K. (2011). Fast speaker diarization using a high-level scripting language. In *Proc. 2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 553–558.

Gravier, G., Adda, G., Paulsson, N., Carré, M., Giraudel, A., and Galibert, O. (2012). The ETAPE corpus for the evaluation of speech-based TV content processing in the french language. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proc. International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Hatch, A. O., Kajarekar, S., and Stolcke, A. (2006). Within-class covariance normalization for SVM-based speaker recognition. In *Proc. of ICSLP*, page 14711474.

Hernandez-Sierra, G., Calvo, J. R., Bonastre, J.-F., and Bousquet, P.-M. (2014). Session compensation using binary speech representation for speaker recognition. *Pattern Recognition Letters*, 49(0):17 – 23.

Hernando, J. and Zelenák, M. (2012). Speaker overlap detection with prosodic features for speaker diarization. *IET signal processing*, 6(8):798–804.

Hernández-Sierra, G., Bonastre, J.-F., and Calvo de Lara, J. (2012). Speaker recognition using a binary representation and specificities models. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, pages 732–739. Springer Berlin Heidelberg.

Huang, R. and Hansen, J. (2006). Advances in unsupervised audio classification and segmentation for the broadcast news and NGSW corpora. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):907–919.

Huang, Y., Vinyals, O., Friedland, G., Muller, C., Mirghafori, N., and Wooters, C. (2007). A fast-match approach for robust, faster than real-time speaker diarization. In *Proc. IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, pages 693–698.

Imseng, D. and Friedland, G. (2009). Robust speaker diarization for short speech recordings. In *Proc. IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, pages 432–437.

Imseng, D. and Friedland, G. (2010). An adaptive initialization method for speaker diarization based on prosodic features. In *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 4946–4949.

Jin, H., Kubala, F., and Schwartz, R. (1997). Automatic speaker clustering. In *Proc. DARPA Speech Recognition Workshop*, pages 108–111.

Kahn, J., Galibert, O., Quintard, L., Carre, M., Giraudel, A., and Joly, P. (2012). A presentation of the REPERE challenge. In *Proc. International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6.

Kenny, P. (2008). Bayesian analysis of speaker diarization with eigenvoice priors. Technical report.

Kenny, P., Ouellet, P., Dehak, N., Gupta, V., and Dumouchel, P. (2008). A study of interspeaker variability in speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5):980–988.

Kenny, P., Reynolds, D., and Castaldo, F. (2010). Diarization of telephone conversations using factor analysis. *IEEE Journal of Selected Topics in Signal Processing*, 4(6):1059–1070.

Khoa, P. C. (2012). Noise robust voice activity detection. Master's thesis, School of Computer Engineering, Nanyang Technological University, Singapore.

Kinnunen, T. and Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12 – 40.

Lathoud, G. and McCowan, I. (2003). Location based speaker segmentation. In *Proc. International Conference on Multimedia and Expo (ICME)*, volume 3, pages III–621–4 vol.3.

Liu, D. and Kubala, F. (1999). Fast speaker change detection for broadcast news transcription and indexing. In *EUROSPEECH*. ISCA.

Mami, Y. and Charlet, D. (2003). Speaker identification by anchor models with PCA/LDA post-processing. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I–180–I–183 vol.1.

Meignier, S., Bonastre, J.-F., and Igounet, S. (2001). E-HMM approach for learning and adapting sound models for speaker indexing. In *ODYSSEY*, pages 175–180.

Merlin, T., françois Bonastre, J., and Fredouille, C. (1999). Non directly acoustic process for costless speaker recognition and indexation.

Nguyen, P. (2003). Swamp: An isometric frontend for speaker clustering. In *NIST 2003 Rich Transcription Workshop*, Boston, USA.

Nguyen, T. H., Chng, E., and Li, H. (2008). T-test distance and clustering criterion for speaker diarization. In *Proc. INTERSPEECH*.

NIST (2000). 2000 Speaker Recognition Evaluation Evaluation Plan. `http://www.itl.nist.gov/iad/mig/tests/spk/2000/spk-2000-plan-v1.0.htm`. Accessed: 2015-03-06.

Nwe, T. L., Sun, H., Li, H., and Rahardja, S. (2009). Speaker diarization in meeting audio. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4073–4076.

Otterson, S. and Ostendorf, M. (2007). Efficient use of overlap information in speaker diarization. In *Proc. IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, pages 683–686.

Ouellet, P., Boulianne, G., and Kenny, P. (2005). Flavors of gaussian warping. In *Proc. INTERSPEECH*, pages 2957–2960.

Pardo, J., Anguera, X., and Wooters, C. (2007). Speaker diarization for multiple-distant-microphone meetings using several sources of information. *IEEE Transactions on Computers*, 56(9):1212–1224.

Pardo, J. M., Anguera, X., and Wooters, C. (2006). Speaker diarization for multiple distant microphone meetings: mixing acoustic features and inter-channel time differences. In *Proc. INTERSPEECH*, Pittsburgh, PA, USA.

Pelecanos, J. and Sridharan, S. (2001). Feature warping for robust speaker verification. In *2001: A Speaker Odyssey - The Speaker Recognition Workshop*, pages 213–218, Crete, Greece. International Speech Communication Association (ISCA).

Ramírez, J., Górriz, J. M., and Segura, J. C. (2007). Voice activity detection. fundamentals and speech recognition system robustness. In Grimm, M. and Kroschel, K., editors, *Robust Speech Recognition and Understanding*, pages 1–22. I-TECH Education and Publishing.

Rentzeperis, E., Stergiou, A., Boukis, C., Pnevmatikakis, A., and Polymenakos, L. C. (2006). The 2006 Athens Information Technology speech activity detection and speaker diarization systems. In *Proc. of the Third International Conference on Machine Learning for Multimodal Interaction*, MLMI'06, pages 385–395, Berlin, Heidelberg. Springer-Verlag.

Reynolds, D. and Torres-Carrasquillo, P. (2005). Approaches and applications of audio diarization. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages v/953–v/956 Vol. 5.

Reynolds, D. A. and Torres-Carrasquillo, P. (2004). The MIT Lincoln Laboratory RT-04F diarization systems: Applications to broadcast audio and telephone conversations. In *Proc. Fall 2004 Rich Transcription Workshop (RT-04)*.

Rouvier, M., Dupuy, G., Gay, P., Khoury, E., Merlin, T., and Meignier, S. (2013). An open-source state-of-the-art toolbox for broadcast news diarization. In *Proc. INTERSPEECH*.

Rouvier, M. and Meignier, S. (2012). A global optimization framework for speaker diarization. In *Proc. Odyssey: The Speaker and Language Recognition Workshop, Singapore, June 25-28, 2012*, pages 146–150.

Saon, G., Thomas, S., Soltau, H., Ganapathy, S., and Kingsbury, B. (2013). The IBM speech activity detection system for the DARPA RATS program. In *Proc. INTERSPEECH*, pages 3497–3501.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464.

Schwarz, P. (2009). *Phoneme recognition based on long temporal context*. PhD thesis.

Shriberg, E., Ferrer, L., Kajarekar, S. S., Venkataraman, A., and Stolcke, A. (2005). Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4):455–472.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D. A., and Glass, J. R. (2011). Exploiting intra-conversation variability for speaker diarization. In *Proc. INTERSPEECH*, pages 945–948. ISCA.

Siegler, M. A., Jain, U., Raj, B., and Stern, R. M. (1997). Automatic segmentation, classification and clustering of broadcast news audio. In *Proc. DARPA Speech Recognition Workshop*, pages 97–99.

Silovsky, J. and Prazak, J. (2012). Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4193–4196.

Silovsky, J., Zdansky, J., Nouza, J., Cerva, P., and Prazak, J. (2012). Incorporation of the ASR output in speaker segmentation and clustering within the task of speaker diarization of broadcast streams. In *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 118–123.

Sinha, R., Tranter, S. E., Gales, M. J. F., and Woodland, P. C. (2005). The Cambridge University March 2005 speaker diarisation system. In *Proc. INTERSPEECH*, pages 2437–2440.

Solomonoff, A., Quillen, C., and Campbell, W. M. (2004). Channel compensation for SVM speaker recognition. In *ODYSSEY: The Speaker and Language Recognition Workshop*, pages 57–62.

Stolcke, A., Friedland, G., and Imseng, D. (2010). Leveraging speaker diarization for meeting recognition from distant microphones. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4390–4393.

Temko, A., Macho, D., and Nadeu, C. (2007). Enhanced SVM training for robust speech activity detection. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–1025–IV–1028.

Tran, V.-A., Le, V. B., Barras, C., and Lamel, L. (2011). Comparing multi-stage approaches for cross-show speaker diarization. In *Proc. INTERSPEECH*, pages 1053–1056.

Tranter, S. and Reynolds, D. (2006). An overview of automatic speaker diarization systems. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1557–1565.

Valente, F. and Wellekens, C. J. (2004). Variational Bayesian speaker clustering. In *Proc. Odyssey: The speaker and language recognition workshop*, Toledo, SPAIN.

van Leeuwen, D. A. (2010). Speaker linking in large data sets. In *Odyssey 2010: The Speaker and Language Recognition Workshop*, page 35.

van Leeuwen, D. A. and Konecný, M. (2007). Progress in the AMIDA speaker diarization system for meeting data. In Stiefelhagen, R., Bowers, R., and Fiscus, J. G., editors, *CLEAR*, volume 4625 of *Lecture Notes in Computer Science*, pages 475–483. Springer.

Vaquero, C., Ortega, A., Villalba, J. A., Miguel, A., and Lleida, E. (2010). Confidence measures for speaker segmentation and their relation to speaker verification. In *Proc. INTERSPEECH*, pages 2310–2313. ISCA.

Vijayasenan, D., Valente, F., and Bourlard, H. (2009). An information theoretic approach to speaker diarization of meeting data. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(7):1382–1393.

Wooters, C., Fung, J., Peskin, B., and Anguera, X. (2004). Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system. In *In Proc. RT-04F Workshop.*

Wooters, C. and Huijbregts, M. (2008). The icsi rt07s speaker diarization system. In Stiefelhagen, R., Bowers, R., and Fiscus, J., editors, *Multimodal Technologies for Perception of Humans*, volume 4625 of *Lecture Notes in Computer Science*, pages 509–519. Springer Berlin Heidelberg.

Yamaguchi, M., Yamashita, M., and Matsunaga, S. (2005). Spectral cross-correlation features for audio indexing of broadcast news and meetings. In *Proc. INTERSPEECH*, pages 613–616.

Yang, Q., Jin, Q., and Schultz, T. (2011). Investigation of cross-show speaker diarization. In *Proc. INTERSPEECH*, pages 2925–2928.

Yella, S. H. and Bourlard, H. (2014). Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(12):1688–1700.

Zhu, X., Barras, C., Lamel, L., and Gauvain, J.-L. (2006). Speaker diarization: From broadcast news to lectures. In *Machine Learning for Multimodal Interaction*, pages 396–406.

Zhu, X., Barras, C., Lamel, L., and Gauvain, J.-L. (2008). Multi-stage speaker diarization for conference and lecture meetings. In Stiefelhagen, R., Bowers, R., and Fiscus, J., editors, *Multimodal Technologies for Perception of Humans*, pages 533–542. Springer-Verlag, Berlin, Heidelberg.