On-line speaker diarization for smart objects

Giovanni Soldi

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0061

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Image »

présentée et soutenue publiquement par

Giovanni SOLDI

le 24 Octobre 2016

On-line speaker diarization for smart objects

Diarisation du locuteur en temps réel pour les objets intelligents

Directeur de thèse : **Prof. Nicholas EVANS, EURECOM**

Prof. John S. D. Mason , Swansea University, UK
Prof. Magne H. Johnsen, NTNU, Norway
Dr. Christophe Beaugeant, Intel, France
Prof. Jean-Luc Dugelay, EURECOM, France

THÈSE

DISSERTATION

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
from TELECOM ParisTech

Specialization: Signal & Image

Giovanni Soldi

On-line speaker diarization for smart objects

Defense scheduled on the 24th of October 2016
before a committee composed of:

Reviewers	**Prof. John S. D. Mason, Swansea University, UK**
	Prof. Magne H. Johnsen, NTNU, Norway
Examiners	**Dr. Christophe Beaugeant, Intel, France**
	Prof. Jean-Luc Dugelay, EURECOM, France
Thesis supervisor	**Prof. Nicholas Evans, EURECOM, France**

The journey, not the destination matters...
— T.S. Eliot

To my mum Eleonora, my family and my lovely Aniko.

Acknowledgments

First of all, I would like to thank my supervisor Nicholas Evans for giving me the opportunity to pursue a PhD degree in a totally new field, for his constant support and to motivate me throughout all the entire PhD. Big thanks are also reserved for Christophe Beaugeant and Intel, for sponsoring my PhD and giving me the opportunity to work in an amazing location like Côte d'Azur. I would also like to thank my Professor from Lund University, Andreas Jakobsson, who initially introduced me to scientific research and convinced me to pursue this long journey.

One great thing of spending many years in a foreign country is all the friends that you get and with whom you share most of the time. Special thanks go to Robin and Rajeev for always letting the doors of their house open, for the huge amount of Indian dinners that saved me from my empty fridge and for all the time spent together, to Jose El Tiburon, for always welcoming me into his office during my numerous coffee breaks, the rock climbing sessions and the numerous adventures and Leela to be patient enough to share the office with me for three years and a half. Big thanks for Max, Hector, Pepe and Pramod for enlarging and bringing new ideas into the Speech group and for the numerous evenings and moments accompanied by live music. Beside the usual working and leisure moments, I really appreciated your constant support and encouragement even in the toughest moments of my life. I would also like to thank my other friends and colleagues who are still at Eurecom or I had the chance to meet them for shorter periods, Ester, Rui, Valeria, Chiara, Raj, Federico Alegre and Claudiu. Special thanks go to the friends outside EURECOM, in primis, Federico and Aldo.

Besides the big achievement of obtaining a PhD degree, these three years and the half have represented also an amazing experience in many other aspects. Living in this region has given me the opportunity to pursue my goal of practising constantly scuba diving, increase my experience and knowledge in this discipline and become a scuba

diving instructor. Big thanks go to the people of Pianeta Blu with whom I spent lot of weekends learning and helping at the diving center in Ventimiglia.

To conclude, special thanks go to my family, my sister Eliana, Davide, aunt Gigia and uncle Giangi who always supported me throughout all my educational path and in all my choices and especially my mum Eleonora. Without your teachings, your endless love and enormous support, your continuous motivation, your advices and your constant presence, all this would have never been possible. Last but not least, I would like to thank my lovely girlfriend Aniko, for her constant support and for having shared with me these exciting years of my life. Love you!

Abstract

Speaker diarization aims to detect “who spoke and when" in a given audio stream. Different applications, such as document structuring or information retrieval have driven the exploration in many diverse spheres, from broadcast news to lectures, phone conversations and meetings.

Nowadays, almost all current diarization systems are off-line and ill-suited to the growing need for on-line or real-time diarization, stemming from the increasing popularity of powerful, mobile smart devices, the increasing interest in the Internet of Things (IoT), the diffusion of always listening sensors and the increasing demand of speech-based context recognition applications. While a small number of such systems have been reported, the majority has focused on less challenging domains, such as broadcast news and plenary speeches characterised by long speaker turns and low spontaneity.

The first part of this thesis entails the problem of on-line speaker diarization. A completely unsupervised adaptive on-line diarization system for challenging and highly spontaneous meeting data is proposed. While not dissimilar to those previously reported for less difficult domains, high diarization error rates illustrate the challenge involved due to the initialisation of the speaker models with little amount of speech data.

To overcome this problem, a semi-supervised approach to on-line diarization whereby speaker models are seeded with a modest amount of manually labelled data is proposed. In practical applications, such as meetings, such data can be obtained readily from brief round-table introductions. On-line speaker modelling is also improved by applying an efficient incremental maximum a-posteriori adaptation (MAP) procedure. The resulting system outperforms a baseline, off-line system by a significant margin and when configured appropriately, error rates may be low enough to support practical applications.

The second part of this dissertation addresses instead the problem of phonetic normalisation when dealing with short-duration speaker modelling. Similarly to many automatic speech processing applications, for instance automatic speaker verification (ASV), on-line diarization requires the training and learning of speaker models with scarce quantity of speech data. Performance can degrade significantly in the face of phonetic variation, which is not marginalised. In this regard, phone adaptive training (PAT) is a recently proposed technique which aims to derive a new acoustic feature space in which the influence of phone variation is minimised while that of speaker variation is maximised.

First, PAT is assessed and optimised at the speaker modelling level and in the context of automatic speaker verification (ASV). Experiments, performed under strictly controlled conditions and using manually derived phonetic transcriptions, show that PAT improves the performance of a state-of-art iVector ASV system by 50% relative to the baseline. Then, PAT is further developed towards a completely unsupervised system using automatically generated acoustic class transcriptions whose number is controlled by regression tree analysis. It is shown that PAT still delivers significant improvements in the performance of a state-of-the-art iVector ASV system even when accurate phonetic transcriptions are not available. Finally, a first attempt at combining PAT and semi-supervised on-line diarization using the TIMIT database confirms the potential of PAT in improving real-time speaker modelling and motivates further research in this particular direction.

Table of contents

List of figures

List of tables

Acronyms

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first appears in the text.

AHC	Agglomerative Hierarchical Clustering
ASV	Automatic Speaker Verification
BIC	Bayesian Information Criterion
cMLLR	Constrained Maximum Likelihood Linear Regression
DCT	Discrete cosine Transform
DER	Diarization Error Rate
DET	Detection Error Trade-off
DFT	Discrete Fourier Transform
DHC	Divisive Hierarchical Clustering
EER	Equal Error Rate
EM	Expectation Maximisation
GLR	Generalized Likelihood Ratio
GMM	Gaussian Mixture Model
HDP	Hierarchical Dirichlet Process
HMM	Hidden Markov Model
IB	Information Bottleneck
ICR	Information Change Rate
ILP	Integer Linear Programming
IoT	Internet of Things
KBM	Binary-key Background Model
LDA	Linear Discriminant Analysis
LFCC	Linear frequency Cepstral Coeffiecient

MAP Maximum a-posteriori adaptation
MAP Maximum a-posteriori adaptation
MDM Multiple Distant Microphones
MFCC Mel-frequency Cepstral Coefficient
MLLR Maximum Likelihood Linear Regression
PAT Phone Adaptive Training
PCA Principal Component Analysis
PLDA Probabilistic Linear Discriminant Analysis
RT Rich Transcriptions
SAD Speech Activity Detection
SAT Speaker adaptive training
SDM Single Distant Microphone
UBM Universal Background Model

Chapter 1

Introduction

In recent years there has been an increasing interest in the Internet of Things (IoT). IoT represents a network of physical devices, vehicles, buildings and other items which are embedded with electronics, software, sensors that enable these objects to communicate, collect and exchange data among one another. "Things" in the IoT sense, can refer to a wide variety of devices for all different contexts: home automation and security, workplace productivity, safe driving, home entertainment such as smart TVs, smart fridges, hands free loud speakers, health and fitness.

Due to the ubiquity of connected smart devices equipped with multiple sensors, applications that exploit all the collected information and data to provide personalised services depending on the user context have always undergone evolutionary advances. Context awareness refers to the process of automatically identifying the context around a smart device. Devices equipped with different sensors may have information about the circumstances under which they are able to operate and based on rules provide improved services to the user needs. Many sources of information for context sensing are available, such as GPS, WiFi antennas, luminance, acceleration, or temperature. Nevertheless, audio and speech provide a rich source of context-related information and there already exists suitable sensors, i.e., microphones, in many portable devices.

Nowadays, multiple smart objects, for instance smart-phones and smart TVs, are able to recognise the voice of a single person after giving a reasonable amount of training data and provide personalised services according to different commands and user preferences. However, a group of people, such as a family in a private house or workers in an office environment, might have the need to interact with different smart objects. In this case, it becomes important to determine in real-time "who is speaking

now?" in order to supply an interactive service that caters to the needs of the person who is currently speaking.

The task of determining the speakers involved in a given audio stream and their corresponding intervals of activity is known as the task of **speaker diarization**.

1.1 The speaker diarization task

The task of speaker diarization has been formally defined and described by the National Institute of Standards and Technology (NIST) during the Rich Transcription (RT) evaluation campaigns. It entails the segmentation and clustering of an audio stream into homogeneous segments based on speaker identities in order to answer the question "**who spoke when?**". In particular, it detects speech changes, corresponding to speaker turns, and using a common label identifies the segments of speech which correspond to the same speaker. Speaker diarization is usually unsupervised and it cannot exploit any a-priori information regarding the involved speakers.

Originally, the main objective of the NIST RT campaign was to enrich the transcriptions of automatic speech recognition (ASR) systems with metadata in order to make them more readable. In addition, speaker diarization has been exploited as an enabling technology relevant to a wide variety of tasks, such as audio indexation, content structuring, navigation, information retrieval and copyright detection.

Speaker diarization has been studied in three different application domains:

- **Phone conversations** where the audio recordings correspond to the oral conversations of two or more speakers through a telephone,
- **broadcast news** where the examined audio recordings correspond to news information from television or radio transmissions. These transmissions are generally characterized by the intervention of different speakers and by an important acoustic variability related to the conditions when the audio was recorded (studio, telephone, noisy environment, music presence, commercial advertisements),
- **meeting recordings** mainly characterised by the presence of more speakers that can communicate from different places, by means of different microphones. The high percentage of overlap, phonetic and channel variability, spontaneity and fast speaker turns characterize this kind of audio recordings.

Meeting recordings, in which multiple speakers are present and participate actively with fast speaker change turns and high percentage of overlap, represent by far the most challenging available data for the speaker diarization task.

Although, speaker diarization has undergone significant advances, partly spearheaded by the international NIST evaluations, the state-of-the-art in speaker diarization has largely evolved around off-line systems, i.e. where an audio stream is processed in its entirety before any segments are assigned speaker labels. Off-line speaker diarization processes the entire audio stream more times if needed in order to detect and describe the involved speakers by means of mathematical models.

However, driven by the expansion of IoT, smart devices and the growing need of speech-based context recognition applications, on-line speaker diarization has attracted increasing interest.

1.2 On-line speaker diarization

Unlike off-line speaker diarization, on-line speaker diarization aims to answer the question **"who is speaking now?"** by determining which speaker is currently active and its interval of activity.

Due to their high computational complexity and latency, the existing state-of-the-art off-line diarization techniques are not easily adapted to face the challenges required by real-time processing. Moreover, even if in recent years, a small number of on-line diarization systems have been reported, the majority focused on applications involving plenary speeches and broadcast news, i.e. [1–3], where the speaker turns are longer and there is less chance of overlapping speech. Moreover, their performances are generally still far from that of typical off-line diarization systems. There is therefore the need to develop on-line diarization systems suitable to support new emerging real- time practical applications, driven by the spread of IoT and smart devices. Since there are still no databases of recordings addressing these emerging applications, meeting recordings are still the most suitable to develop an on-line diarization system due to their spontaneity and variability.

On-line speaker diarization represents a much more challenging task than off-line speaker diarization. Decisions, initialisation and update of speaker models have to be made in real-time and based on short-duration speech segments in order to support

practical applications with an acceptable latency. Longer speech segment durations would provide more robust speaker models at the cost of a higher system latency and at the risk of including impurities, i.e. more speakers in the same speech segment.

1.3 Robust speaker modelling against phonetic variation

Many automatic speech processing applications are required to operate in the face of varying data quantities. When data is plentiful, phonetic or nuisance variation can be implicitly normalised and often has limited or no impact on performance. In contrast, when training data is scarce, then performance can degrade drastically if the phonetic variation is dissimilar to that encountered in testing; for example if during training phase only a set of phones is encountered phonetic variation is no longer marginalised.

Speaker diarization and in particular on-line speaker diarization are two such examples in which speaker models can be initialised and updated with steadily amassed short speech segments or well-trained models can be compared to short test segments. When speaker models are initialised using short-duration speech segments, the speaker models are biased towards the limited phonetic information contained in the short speech segment. Consequently, when speaker models are compared to following short-duration speech segments, misclassification errors are committed due to the differing phonetic content.

This problem strongly requires the development of speaker discriminative modelling techniques in order to isolate the relevant information related to the speakers while marginalising the phonetic content in short-duration speech segments. Research in this direction would allow the training and initialisation of more discriminative speaker models with less amount of speech data and consequently would contribute to the reduction in latency and improve the performance of a typical on-line diarization system. Phone adaptive training (PAT), recently introduced by Bozonnet et al. [4], is a technique meant to create a more discriminative acoustic space in which the phonetic variation is marginalised while the speaker information is retained.

1.4 Objectives and outline of this thesis

The first objective of this dissertation concerns the problem of on-line diarization involving more than two speakers. The goal is to develop an on-line speaker diarization system suitable to support real-time practical applications required by the rapid development of IoT applications and the higher request of speech-based context recognition applications. An unsupervised and semi-supervised on-line diarization system is presented in this thesis.

The second objective of this dissertation consists of the optimization and development of PAT and its potential application to on-line speaker diarization.

An outline of this thesis along with a brief summary of the contributions of each chapter is provided below.

Chapter 2 - State-of-the-art

In this chapter an overview of the most recent research in the topic of speaker diarization is provided. The most spread off-line diarization systems, categorised according to different speaker modelling techniques, are first presented. The last part of this chapter covers few on-line diarization systems presented in literature.

Chapter 3 - Databases and metric

This chapter provides a description of the diarization error rate (DER) metric used to measure the performance of on-line speaker diarization, a description of the main NIST RT evaluation datasets used for on-line speaker diarization experiments and the off-line baseline diarization system used as reference. Finally, it describes the TIMIT database, manually transcribed at the phonetic level used for the development and optimisation of PAT.

Chapter 4 - Unsupervised on-line diarization

While a small number of on-line diarization systems have been previously reported in the literature, truly on-line diarization systems for challenging and highly spontaneous meeting data have not yet been presented. The main contribution of this chapter consists of our first attempt to develop an adaptive, completely un-supervised on-line speaker diarization system for meeting data. The developed system is based on the

sequential introduction and adaptation of speaker models by means of a sequential adaptation procedure. The performance of the system is assessed through experiments in which different segment durations and different speaker model complexities are used. Performance is also assessed in terms of dynamic convergence of the speaker models during time. While the performance is not so dissimilar to that of other systems presented in literature on less challenging domains, high diarization error rates illustrate the challenge ahead.

Part of the work in this chapter has resulted in the following publication:

- *Soldi, G., Beaugeant, C., Evans, N. "Adaptive and online speaker diarization for meeting data" in Proc. European Signal Processing Conf. (EUSIPCO), 2015, Nice, France.*

Chapter 5 - Semi-supervised on-line diarization

After identifying the main bottleneck of unsupervised on-line diarization in the unsupervised initialisation and adaptation of speaker models with short-duration speech segments, the first contribution of this thesis is to investigate a semi-supervised approach to on-line diarization whereby speaker models are seeded off-line with a modest amount of manually labelled data. In practical applications involving meetings or when interacting with smart connected objects, such data can be readily obtained from brief introductions. The question that this chapter tries to address is: what amount of labelled training data is needed to match or overtake the performance of a state-of-the-art off- line baseline diarization system?

The second contribution of this chapter relates instead to an incremental approach to on-line model adaptation which proves instrumental in delivering low diarization error rates. It is shown that such a system can outperform an off-line diarization system with just 3 seconds of speaker seed data and 3 seconds of latency when using an incremental MAP adaptation procedure. By using greater quantities of seed data or by allowing greater latency, then a diarization error rate in the order of 10% can be achieved.

Part of the work in this chapter has resulted in the following publication:

- *Soldi, G., Todisco, M., Delgado, H., Beaugeant, C., Evans, N. "Semi-supervised on-line speaker diarization for meeting data with incremental maximum a-posteriori*

adaptation“ in Odyssey - The Speaker and Language Recognition Workshop, June 2016, Bilbao, Spain.

Chapter 6 - Phone adaptive training

Based on constrained maximum likelihood linear regression (cMLLR), a model adaptation technique, and previous work in speaker adaptive training (SAT) for automatic speech recognition, PAT learns a set of transforms which project features into a new phone normalised but speaker-discriminative space. Originally investigated in the context of speaker diarization, in the first part of this chapter PAT is assessed and optimised at the level of speaker modelling and in the context of automatic speaker verification (ASV) under strictly controlled conditions, including the use of manually derived phone transcripts. PAT performance is analysed when applied to short-duration text-independent ASV as a function of model complexity and for varying quantities of training data, using the TIMIT dataset which is manually labelled at the phone level. It is shown that PAT is successful in reducing phone bias and it improves significantly the performance of both traditional GMM-UBM and iVector-PLDA ASV systems in the case of short-duration training. Moreover, PAT delivers better performance for lower models complexities.

The second part of this chapter instead reports on our efforts to develop PAT into a fully unsupervised system. Contributions include an approach to automatic acoustic class transcription using regression tree analysis. Similarly to the first work, the performance of PAT is analysed in the context of ASV as a function of model complexity and for varying quantities of training data. Experiments show that PAT performs well even when the number of acoustic classes is reduced well below the number of phones thus minimising the need of accurate ground-truth phonetic transcriptions.

Part of the work in this chapter has resulted in the following publications:

- *Soldi, G., Bozonnet, S., Alegre, F., Beaugeant, C., Evans, N. “Short-duration speaker modelling with phone adaptive training“, in Odyssey - The Speaker and Language Recognition Workshop, 2014, Joensuu, Finland.*

- *Soldi, G., Bozonnet, S., Beaugeant, C., Evans, N. “Phone adaptive training for short- duration speaker verification“ in Proc. European Signal Processing Conf. (EUSIPCO), 2015, Nice, France.*

Chapter 7 - PAT for on-line diarization: a first attempt

This chapter presents a first attempt to use PAT with the aim of improving short-duration speaker-modelling in on-line diarization. Due to the unavailability of datasets transcribed at the phonetic level, multi-speaker audio conversations are simulated by joining different sentences from the TIMIT dataset. The semi-supervised on-line diarization system presented in Chapter 6 is used to perform the diarization process. During the off-line phase phone-normalised speaker models are trained by transforming the acoustic features with PAT transforms previously trained. Similarly, in the on-line classification phase, the acoustic features of the incoming speech segments are phone-normalised and classified against the phone-normalised speaker models. Although experiments are performed on simulated data, obtained results confirm the potential of PAT for on-line diarization in providing better results with a lower speaker model complexity and a lower amount of training data, therefore motivating future research in this particular direction.

Chapter 8 - Conclusions and future work

Finally, this chapter concludes the dissertation and points out some possible future research problems.

Chapter 2

State of the art

This chapter provides a brief overview of some aspects of speaker diarization techniques that the thesis aims to cover. For detailed and deep reviews of the current state-of-the-art speaker diarization techniques, we refer the reader to the works in [5] and [6].

First, the task of speaker diarization is presented in detail and then followed by a brief description of main blocks in a typical diarization system. Feature extraction methods and mathematical techniques for speaker modelling are presented in sections 2.2 and 2.4, respectively. A classification of different diarization systems according to the type of algorithm used for the segmentation and clustering stage is given from section 2.5 to section 2.9; Finally, section 2.10 deals with on-line speaker diarization, which constitutes the main topic of this thesis.

2.1 Speaker diarization

As illustrated in Figure 2.1, speaker diarization aims to answer the question **"who spoke when?"** and involves the detection of speaker turns within an audio stream of maximum duration T in order to determine an optimised segmentation $\tilde{G}$ and the grouping together of all same-speaker segments (clustering) in order to obtain an optimised speaker sequence $\tilde{S}$.

Although, different approaches have been presented in the literature to perform speaker diarization, most of them follow the general structure presented in Figure 2.2.

All the blocks are described in the following:

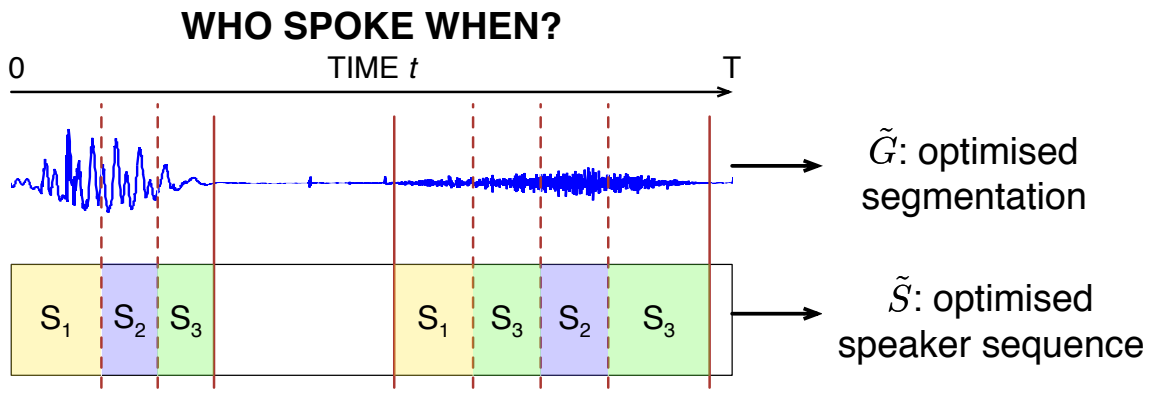


Fig. 2.1 An illustration of the speaker diarization task.

- **Feature extraction:** the audio is parametrized through a set of acoustic features $\boldsymbol{O}$ extracted from the audio signal. The extraction method should be chosen in order to have features that are discriminant with respect to the speakers in the audio.
- **Speech Activity Detection (SAD):** during this step the audio parametrized by the acoustic features is segmented in order to detect the boundaries of pure speech and non-speech regions (music, noise, silence).
- **Segmentation:** once the acoustic features and the speech segments are obtained the signal is segmented into acoustically homogeneous regions. The segmentation aims at finding the boundaries between different and unknown speakers. An initial segmentation usually aims at creating initial speech segments where in each segment is supposed to be present only one speaker.
- **Clustering:** the clustering stage aims at grouping and merging together the initial speech segments, obtained from the segmentation step, that are mostly similar according to predefined metrics and are supposed to contain the same speaker. The speech segments belonging to the same cluster are identified with a unique cluster identifier referring to a unique speaker. The set of start and end times obtained from the segmentation step together with the labels of the clustering stage provide the final output of the diarization system.

Segmentation and clustering are inter-dependent stages and especially in off-line speaker diarization systems can be repeated iteratively one after each other in order to refine the diarization output. The output of the clustering stage corresponding to the

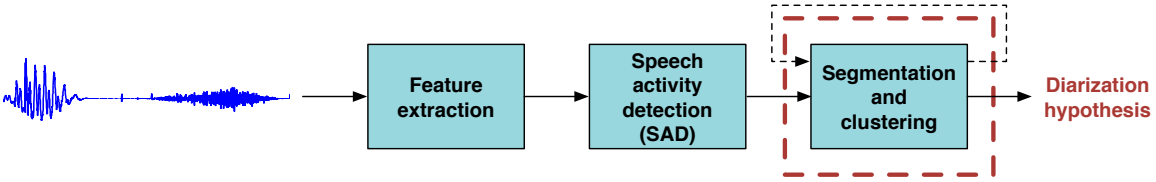


Fig. 2.2 General structure of a speaker diarization system.

speakers present in the audio could be given as an input to the segmentation stage to refine the boundaries of the speaker segments and the refined segmentation stage could be fed into the clustering stage until an optimal solution is obtained. The clustering and segmentation steps could also be followed by a re-segmentation aiming at refining the boundaries of the obtained speaker clusters.

Mathematically, the problem of speaker diarization can be formulated in the following way:

$$\left(\tilde{S}, \tilde{G}, \tilde{\Delta}\right) = \underset{S,G,\Delta:S\in\Gamma(\Delta)}{\arg\max} P(S, G|\mathbf{O}), \tag{2.1}$$

$$\approx \underset{S,G,\Delta:S\in\Gamma(\Delta)}{\arg\max} P(\mathbf{O}|G, S), \tag{2.2}$$

where $\tilde{\Delta}$ represents an optimised speaker inventory, $\tilde{S}$ and $\tilde{G}$ represent an optimised speaker sequence and segmentation respectively, $\Gamma(\Delta)$ is the set of possible speaker sequences and $\mathbf{O}$ is the set of acoustic features.

The detection of silence, background noises, music, acoustic events or more generally non-speech events are usually needed as a preliminary step to perform actual speaker diarization, and it is usually included in most of speaker diarization systems. However, the algorithms needed to detect these acoustic events usually differ from the ones needed to detect and cluster speakers speech segments. In this thesis, we thus focus on techniques used to classify different speakers in an audio signal rather than speech/non-speech or other acoustic events.

2.2 Features

In order to segment and cluster the audio into portions corresponding to different speakers, the audio signal must be parametrised by acoustic features that are able to separate and highlight speaker discriminative characteristics. The most popular and

common features used for the task of speaker diarization are Linear Frequency Cepstral Coefficients (LFCC) and Mel Frequency Cepstral Coefficients (MFCC). The main objective of LFCC and MFCC features is to model the vocal tract of the speakers while separating it from the pitch. Figure 2.3 illustrates the procedure for the extraction of LFCC features given an audio stream. The audio is processed with moving overlapping windows of a fixed maximum size. For each window, the log-magnitude of the Fourier transform of the audio signal in the window is computed. Then, the inverse Fourier transform is re-computed and only the first N coefficients, which represent the vocal tract, are retained.

Figure 2.4 illustrates instead the procedure to extract the MFCC features from an audio signal. The main difference with the LFCC features is that the power spectrum obtained after the Fourier transform is mapped above onto the mel-scale, using triangular overlapping windows. The logs of the powers at each of the mel-frequencies are then taken and the discrete cosine transform is applied to them rather than the inverse Fourier transform.

Both MFCC and LFCC parametrisations are known to obtain state-of-the art performance in speaker diarization. However, the fact that these features have been employed successfully also in the speech recognition task, where the speaker information is less relevant, has motivated researchers to explore more speaker discriminant features.

In [7] a set of features composed by energy, pitch frequency, peak-frequency centroid, peak-frequency bandwidth, temporal feature stability of the power spectra, spectral shape and white noise similarities is used for segmenting the audio into different classes, including speech, silence, noise and crosstalk. The same features are as well used to identify speakers in the obtained speech segments.

MFCC and LFCC are short-term features, meaning that they are extracted from short windows of time. Other prosodic and long-term features have been proposed in literature with the aim of improving speaker characterization [8]. In [9] Friedland et al. studied a total of seventy features including prosodic and long-term features. Although it is shown that certain statistics extracted from the pitch and formant estimation could be used together with MFCC features in order to slightly improve the performance of speaker diarization, the gain was not significant enough to take these features into consideration.

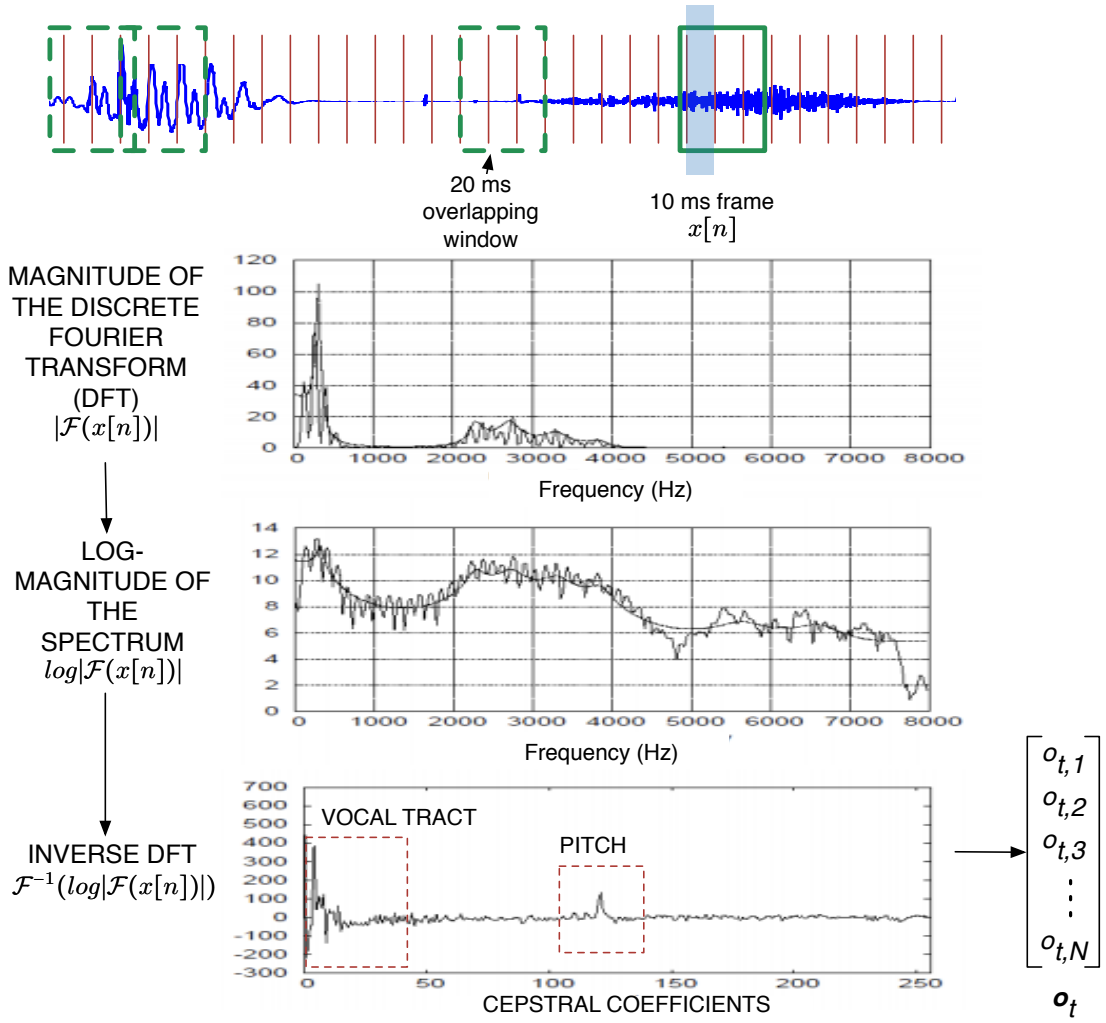


Fig. 2.3 An illustration of how Linear Frequency Cepstral Coefficients (LFCC) are computed.

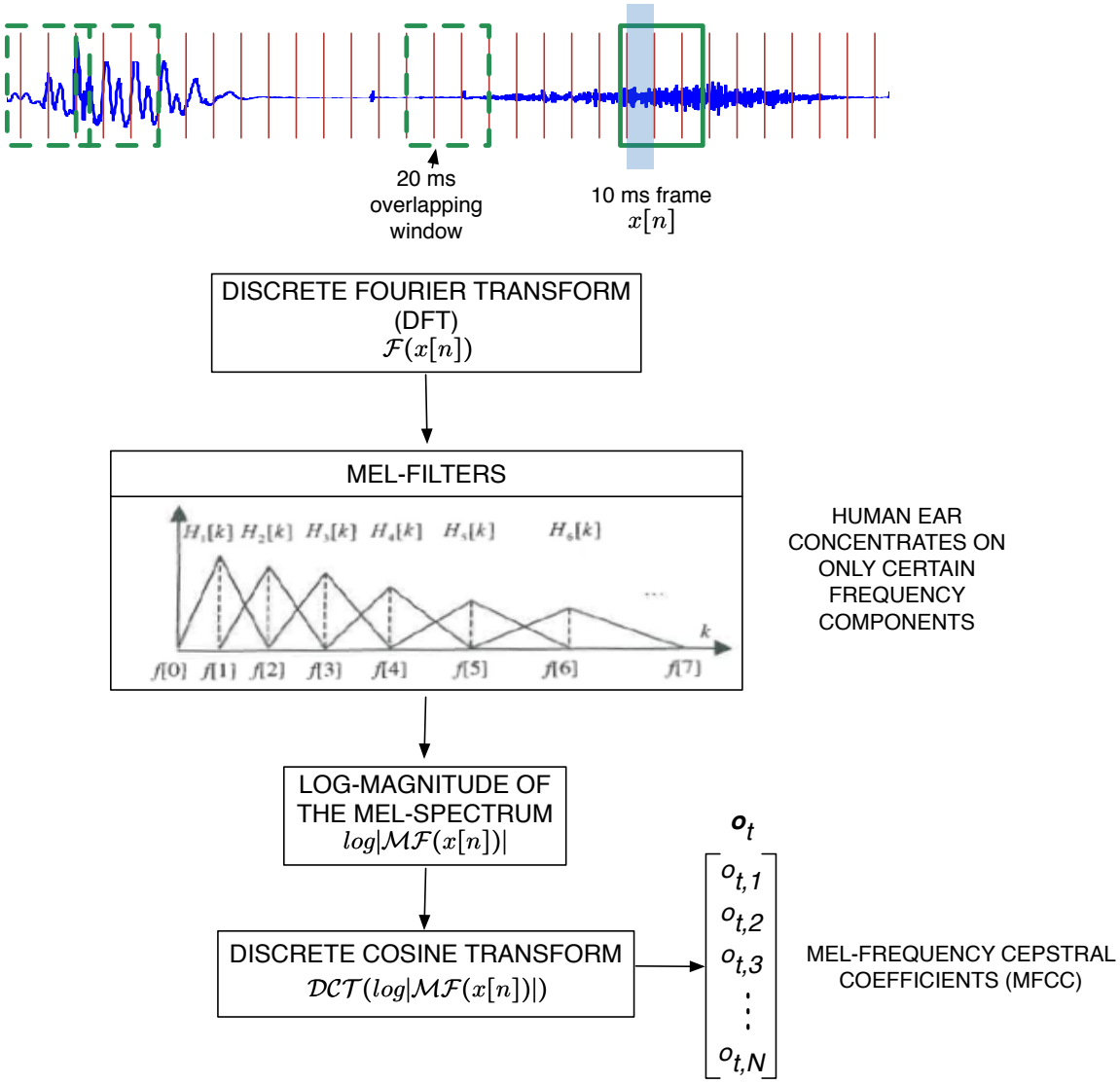


Fig. 2.4 An illustration of how Mel Frequency Cepstral Coefficients (MFCC) are computed.

Feature normalisation techniques, such as feature warping [10] have been also explored in order to reduce channel variability and background noise. Feature warping was applied successfully in [11] and [12]. Despite this, Kenny et al. have shown in [13] that the performance of a speaker diarization system for telephone conversations is better when non-normalised features are used.

Other features not directly related to acoustic parameters of the speakers present in a given audio signal have shown to help speaker diarization under certain conditions. In environments where more than one microphone is available to capture the audio signal, the time-delays between microphones, which are related to the position of the speakers, have shown to improve speaker diarization performance, as far as the speakers remain static [14].

2.3 Speech/non-speech detection

The main scope of speech activity detection (SAD) system is to identify speech and non-speech regions in an audio stream. An efficient SAD is critical for the performance of a speaker diarization system as the errors performed during this process will contribute in two different ways to the output of a diarization system: missed-speech errors and false-speech errors. The missed speech errors are caused by excluding speech regions and by classifying them as non-speech, thus providing less speech data useful to model the speaker clusters and thus resulting in poor speaker clustering. On the contrary, false-speech errors are caused by mistakenly classifying non-speech regions as speech regions, thus introducing impurities when modelling the speaker clusters. It is thus evident, the need of an efficient SAD system as a preprocessing step before the segmentation and clustering stage.

The problem of SAD has been largely studied in various situations such as speech enhancement, recognition and coding [15] . The main algorithms can be classified into three categories: energy-based, model-based and hybrid approaches which are listed in the following.

2.3.1 Energy-based SAD algorithms

Energy-based SAD algorithms detect speech/non-speech regions according to thresholds on the short-term spectral energy [16, 17]. These methods do not need any type of

labelled data and are mostly employed in telephone speech conversations. However, in scenarios with a variety of acoustic events such as noise, music, channel variation, e.g. meeting conversations or broadcast news, characterized by high energy-levels, these methods are not able to provide satisfying performance [18, 19].

2.3.2 Model-based SAD algorithms

Model-based SAD algorithms use pre-trained Gaussian Mixture models (GMM) on previously labelled speech and non-speech data in order to identify classes within non-labelled data [20]. The pre-trained GMM models can also be adapted iteratively to the test data [21]. Usually, GMM models are trained for both speech and non-speech classes and acoustic features in the test data are iteratively realigned to the models through Viterbi decoding. A minimum speech segment duration can be applied in order to avoid extremely short-duration speech segments. Transitions between speech and non-speech models could also be modelled with the aid of a two-state HMM model. In some cases, different GMM models can be trained in order to model other acoustic events such as music, noise, speech overlapped to music, speech overlapped to noise, silence. Gender and channel dependent models have been proposed in other works in order to improve the SAD output. The main drawback of using multiple classes is the need of sufficient training data to train all the models and generalisability to new environments. Other discriminant classifiers based on Linear Discriminant Analysis (LDA) [22], Support Vector Machines [23] and multi-layer perceptrons (MLPs) [24] have also been proposed in literature. In the latter case, the output layer is used to obtain class posterior estimates as scaled likelihoods in the Viterbi decoding process to perform SAD.

2.3.3 Hybrid SAD algorithms

In order to avoid the issue of generalisability of previously trained models to new data and alleviate the need of labelled training data, hybrid approaches to SAD have been proposed [16, 25]. These methods combine both an energy-based SAD algorithm and a model-based SAD algorithm to perform speech/non-speech detection. Initially, an energy-based SAD is used to detect silence regions in an totally unlabelled audio. Then, the labelled speech regions with highest confidence are used to train new speech/non-

speech models or adapt existing ones that will be later used with a speech/non-speech model-based SAD system.

In the next section, we will present the main mathematical techniques that are used to model the speaker clusters during the segmentation and clustering stage.

2.4 Speaker modelling techniques

Speaker diarization involves the mathematical modelling of speakers in the audio, in particular during the segmentation and clustering stage. Gaussian Mixture Models (GMM) have been largely explored in speaker diarization and are mostly used to model the variability of speech and speakers. Hidden Markov Models (HMM) have been used together with GMM models to model the transitions among speakers. In recent years, more sophisticated techniques such as Joint Factor Analysis and Total Variability methods, that have become the state-of-the-art in speaker verification, have been successfully applied to speaker diarization.

2.4.1 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a generative model widely used in speaker diarization as well as speaker verification. It is a semi-parametric probabilistic method that offers the advantage of adequately representing speech signal variability. The distribution of a GMM model λ modelling D-dimensional acoustic feature vectors is given by a convex combination of a fixed number K of Gaussian distributed components. The likelihood of observing an acoustic feature vector $\boldsymbol{o}$ given this model λ is computed according to the following equation:

$$Pr\left(\boldsymbol{o}|\lambda\right) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{o}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{2.3}$$

where w_k are the components weights such that $\sum_{k=1}^{K} w_k = 1$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean and the covariance matrix of the k-th Gaussian component.

Practical speaker diarization systems use diagonal covariance matrices instead of full covariance matrices to define GMM models. Provided a set of independent acoustic feature vectors $\boldsymbol{O} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_L$, the log-likelihood of the sequence $\boldsymbol{O}$, given the GMM

model λ, is the sum of the log-likelihoods of each feature vector $\boldsymbol{o}_l$ given that model. The corresponding log-likelihood is thus:

$$logPr\left(\boldsymbol{O}|\lambda\right) = \sum_{l=1}^{L} logPr\left(\boldsymbol{o}_l|\lambda\right) \tag{2.4}$$

The Expectation Maximization (EM) algorithm [26] is used to learn the GMM parameters based on maximization of the expected log-likelihood of the data. In most speaker diarization systems, we do not have enough data to train a speaker-dependent GMM model using the EM algorithm, especially when dealing with short speech segments. To overcome these difficulties, a more general speaker-independent GMM Universal Background Model (UBM) can be trained, under the assumption that this model will adequately describe the underlying characteristics of a large speaker population. Generally, the UBM is trained on a large speech dataset with different number of speakers. The speaker-dependent GMM model is then derived from the UBM model by Maximum A Posteriori (MAP) adaptation, a technique introduced in the next subsection.

2.4.2 MAP adaptation

In most cases data is typically too scarce to warrant reliable EM estimates of speaker-dependent GMM models. In contrast, the generally large amounts of data used in estimating the speaker-independent UBM allows this model parameters to serve as an appropriate starting point to derive speaker-dependent models.

Accordingly, the parameters of a speaker-dependent models are determined via Maximum A Posteriori (MAP) adaptation [27] of the initial parameters of the prior model (UBM), using the available speaker-dependent acoustic features. By virtue of the typically limited amount of corresponding data, the resulting MAP adapted parameters will tend to be much more reliable than their maximum likelihood trained counterparts (EM-algorithm).

Since typical on-line diarization has to deal with the on-line learning and adaptation of speaker models with short-duration speech segments, MAP adaptation constitutes a fundamental wheel of the unsupervised and semi-supervised systems proposed in Chapter 4 and 5, respectively. Conventional off-line MAP adaptation is described in detail in Chapter 4, Section 4.1.1. Derived sequential and incremental versions,

optimised for on-line diarization, are instead described in Chapter 4, Section 4.1.2 and Chapter 5, Section 5.2.1 respectively.

2.4.3 Joint-Factor analysis

Joint-Factor analysis approaches are based on the idea that a speaker and channel dependent GMM model obtained by MAP adaptation of a UBM model of a number K of Gaussian components and parametrized by a feature space of dimension D, can also be interpreted as a single super-vector of dimension KD generated by concatenating all the means of each Gaussian component and a diagonal super covariance matrix of dimension $KD \times KD$ generated by respectively concatenating (on its diagonal) all the diagonal covariances of each Gaussian component.

The main assumption on which the theory of Joint-Factor Analysis lies is that a high-dimensional speaker super-vector can live in a lower-dimensional subspace. In particular, a speaker related super-vector $\boldsymbol{M}$ dependent on a particular channel can be broken into the sum of two components super-vectors as follows:

$$\boldsymbol{M} = \boldsymbol{s} + \boldsymbol{c} \tag{2.5}$$

where super-vector $\boldsymbol{s}$ depends on the speaker and super-vector $\boldsymbol{c}$ depends on the channel.

Moreover, one can write:

$$\boldsymbol{s} = \boldsymbol{m} + \boldsymbol{V} \cdot \boldsymbol{y} + \boldsymbol{F} \cdot \boldsymbol{z} \tag{2.6}$$

$$\boldsymbol{c} = \boldsymbol{U} \cdot \boldsymbol{x} \tag{2.7}$$

where $\boldsymbol{V}$ and $\boldsymbol{U}$ are two-low rank matrices that represent the lower dimensional subspaces in which respectively the speaker and channel variations lie. Lastly, $\boldsymbol{m}$ is the channel and speaker independent super-vector that can also be interpreted as the super-vector derived by the general speech UBM model when stacking the means of all his Gaussian components, while $\boldsymbol{F}$ is a diagonal $KD \times KD$ matrix that serves a purpose similar to that of MAP adaptation. In particular, it models the residual variabilities of a speaker that are not captured by the matrix $\boldsymbol{V}$.

The exact details of this theory are beyond the scope of this thesis, but a thorough explanation can be found in [28]. The terminology Joint Factor Analysis comes from

the fact that there are three latent variables to be estimated ($\boldsymbol{x}$,$\boldsymbol{y}$, and $\boldsymbol{z}$) jointly. Traditional Factor Analysis usually involves only one latent variable.

2.4.4 iVectors-based approaches

The Joint-Factor analysis speaker modelling approach provides powerful tools to model both the speaker and channel variations. However, its intrinsic complexity both in the theoretical and implementation sides motivates for a more simplified solution. The total variability approach suggests to jointly model both the channel and speaker related variabilities by means of a unique lower dimensional subspace.

In particular a speaker and channel dependent super-vector $\boldsymbol{M}$ can be decomposed as:

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T} \cdot \boldsymbol{w} \tag{2.8}$$

where $\boldsymbol{T}$ is a low-rank rectangular matrix, referred to as Total Variability Matrix, that represents the new total variability space and $\boldsymbol{w}$ is a low-dimensional random vector with a Gaussian normal distributed prior $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The components of $\boldsymbol{w}$ are referred to as total factors. We refer to iVector as the posterior expectation of $\boldsymbol{w}$. For some speech utterances u, its associated i-vector $\boldsymbol{w}_u$ can be seen as a low-dimensional summary of the distribution of the acoustic features in the utterance u. iVectors have been employed successfully in the field of speaker verification reaching state-of-the-art performance.

2.4.5 Binary keys approaches

Speaker modelling techniques based on binary keys have been proposed initially in the context of speaker verification, e.g. [29]. This speaker modelling technique is based on the idea of representing a set of acoustic features $\boldsymbol{O} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_N]$, with a unique binary key vector $\boldsymbol{b} = [b(1), \ldots, b(M)], b(i) = \{0, 1\}$. This representation facilitates the comparison among speech utterances as the actual comparisons are reduced to simple binary operations. In order to compute binary key vectors, a generalisation model of a UBM speech model have been introduced: the binary-key background model (KBM model). The KBM model is a derived GMM model that aims to represent all the phonetic and speaker specificities and characteristics. The KBM model is a set of M discriminative Gaussian models with the aim of covering and modelling the

acoustic space of the speakers. Initially, the KBM model was obtained by concatenating the Gaussian components of different speaker models (Anchor Models) from a large database and by choosing the most discriminative Gaussian model according to some metrics. Setting an element $b(i) = 1$ indicates that the i-th Gaussian of the KBM model coexists in the acoustic space of the data being modelled. In order to obtain a binary key, for each acoustic feature vector $\boldsymbol{o}_i$ the N_g Gaussian models of the KBM model with the highest likelihood are selected and stored. Then, the count of how many times each Gaussian model has been selected across all the acoustic feature vectors $\boldsymbol{O}$ is registered in a cumulative vector $\boldsymbol{CV} = [CV(1), \ldots, CV(M)]$. Finally, the binary key is obtained by setting to 1 the positions with the highest value in the cumulative vector $\boldsymbol{CV}$. Intuitively, the binary key keeps the Gaussian components of the KBM model that best model the acoustic space of data.

When, two speech utterances from two different speakers u_1 and u_2 need to be compared, binary keys are computed for both utterances and compared through simple and computationally efficient binary metrics. A simple similarity metric is defined for instance as:

$$S(\boldsymbol{b}_{u_1}, \boldsymbol{b}_{u_2}) = \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{b}_{u_1}(i) \wedge \boldsymbol{b}_{u_2}(i)) \tag{2.9}$$

where $\wedge$ indicates the logic operator AND between any two bits. If the utterances u_1 and u_2 come from the same speakers, the respective binary vectors would share similar Gaussian components, thus obtaining a high similarity score in equation (2.9). On the contrary, when the utterances come from different speakers, the binary vectors do not share the same Gaussian components, thus obtaining a low similarity score in equation (2.9).

This new speaker modelling technique presents several advantages:

- it permits to represent a sample utterance and a reference utterance by compact binary vectors,
- it allows the comparison of a sample utterance and a reference utterance by computational efficient similarity operations, for instance (2.9),
- the speaker modelling is shifted towards the extraction of binary keys provided an utterance u and the KBM model, rather than the training of GMM speaker models.

2.5 Segmentation and clustering

State-of-the-art off-line diarization approaches can be mainly classified according on how the segmentation and clustering stages are performed in two groups: bottom-up and approaches.

- **Bottom-Up:** Bottom-Up hierarchical clustering or agglomerative hierarchical clustering (AHC) methods starts with a large number of speaker clusters or speech segments (the finest partition), each supposed to contain a single speaker and iteratively merge the most similar clusters until a stopping criterion is met. This kind of a technique is the most popular in speaker diarization systems for the fact that is really straight-forward to apply to a set of speech segments output of a speaker segmentation system. Usually a distance matrix, containing the distances within all the possible cluster pairs is computed and the pair of clusters with the lowest distances are merged. The distance matrix is then updated according to the new set of clusters. The whole scenario is repeated iteratively until some stopping criterion is reached, upon which it should ideally remain one cluster per speaker.

- **Top-Down:** Top-down hierarchical clustering or divisive hierarchical clustering (DHC) methods are much less common than the counterpart bottom-up systems in literature. However, recent works have shown that these systems could reach comparable performance and with a less computational effort. Top-Down systems are initialized with a small number of clusters (usually a single one) containing several speech segments from different speakers. The initial clusters are then iteratively split until a stopping criterion is met and the optimal number of speakers is reached.

The main difference between Top-Down and Bottom-Up approaches is visualized in Figure 2.5.

2.6 AHC approaches

AHC approaches (also referred to as bottom-up approaches) are the most common approaches in literature for off-line diarization. Generally they consists of dividing the audio stream in a higher number of initial speech segments considered as pure as

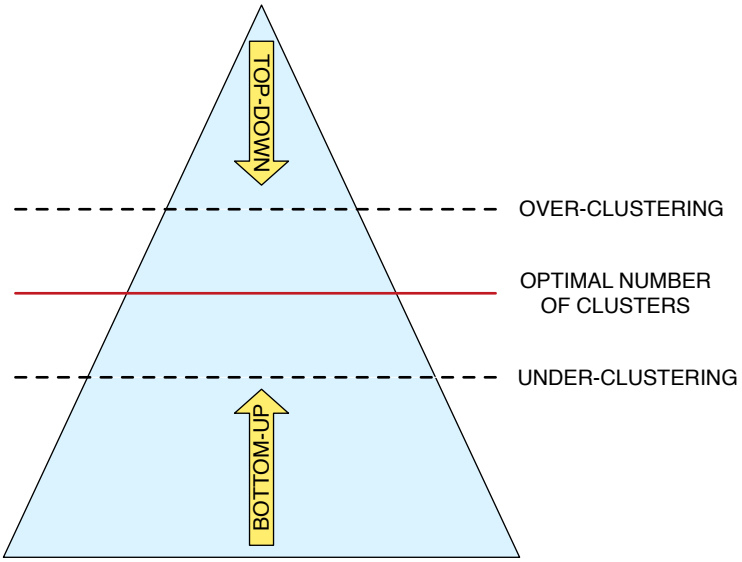


Fig. 2.5 Differences between top-down and bottom-up approaches. Bottom-up methods start from a high number of speaker clusters and continue merging them till reaching the optimal number of clusters. On the opposite, top-down methods start from one speaker cluster modelling the entire audio and divide it iteratively till reaching the optimal number of clusters.

possible and cluster iteratively the segments according to a similarity measure until a stopping criterion is reached. Bottom-up methods can be further distinguished on the speaker modelling technique used in order to model the speaker clusters.

2.6.1 HMM-GMM based approaches

Bottom-up approaches based on HMM-GMM modelling consists of modelling the speakers by means of GMM models with different number of Gaussian components. Transitions between speakers are modelled by means of Hidden Markov Models (HMM). Two systems that are well representative are the ICSI system proposed in [30] and the system proposed by I2R as published in Nguyen et al. [31]. Main difference between the two systems is the distance chosen to determine when two speaker clusters need to be merged. In the ICSI system a Bayesian Information Criterion (BIC) based-metric is used while in the I2R system a Generalized Likelihood Ratio (GLR) metric is used. In the following, the two systems will be briefly described and summarised.

ICSI diarization system

The speech regions output of the SAD process are initially split into a high number of clusters N which exceeds the number of speakers. A Hidden Markov Model is then built with a number of states equal to the number of initial clusters. To each state is then associated a simple Gaussian model trained by EM algorithm. Several iterations of model training and Viterbi alignments are then performed in order to refine the initial segment boundaries and speaker models.

The metric used to define the similarity between speaker clusters is meant to describe the inter-cluster distance and is the ΔBIC distance. Given two speaker clusters C_1 and C_2, the BIC metric aims to compare two hypotheses:

- **H1** hypothesis under which C_1 and C_2 corresponds to two different speakers
- **H2** hypothesis under which C_1 and C_2 corresponds to the same speaker

Mathematically, the ΔBIC distance can be expressed as in the following:

$$\begin{aligned}\Delta BIC =& BIC(H_1) - BIC(H2) \\ =& N_{1,2} log(|\Sigma_{1,2}|) - N_1 log(|\Sigma_1|) - N_2 log(|\Sigma_2|) \\ -& \lambda \frac{1}{2}(d + \frac{1}{2} d(d+1)) log(N_{1,2})\end{aligned} \tag{2.10}$$

where λ is a tunable parameter, N_1 and N_2 are respectively the number of acoustic features for cluster C_1 and C_2, $N_{1,2}$ corresponds simply to the sum of N_1 and N_2, Σ_1, Σ_2 are respectively the covariances of cluster C_1 and C_2, $\Sigma_{1,2}$ is the joint covariance of the joint clusters C_1 and C_2.

The ICSI system utilizes a slightly different version of the BIC metric, as described in [32], where there is no presence of the tunable parameter λ. This is achieved by ensuring that, for any given BIC comparison, the difference between the number of free parameters in the two hypotheses is zero.

During the clustering stage of the diarization system, the ΔBIC distance is calculated for each pair of clusters and the pairs with the highest similarity are merged. After the merging of two clusters, the acoustic features are realigned through Viterbi realignment. Cluster merging and Viterbi realignment are repeated until the similarity between any pair of clusters is lower than a fixed threshold.

I2R diarization system

In the I2R system the speech regions output of the SAD process are split into 30 initial homogeneous clusters and on each of them a GMM model of 4 components is trained. Each cluster is then split into small segments of 500 milliseconds and only the 25% of segments that fits best the corresponding GMM models are considered as classified, while the 75% of the worst-fitting segments are then gradually reassigned to each model through Viterbi realignment and adaptation.

During the clustering phase, performed through agglomerative clustering, the speaker GMM models are retrained for each cluster with 16 Gaussian components. The similarity metric used to compare two speaker clusters is based on the BIC-like Information Change Rate (ICR) distance, defined as a normalized version of the Generalized Likelihood Ratio (GLR).

Given two speaker clusters C_1 and C_2, the metric is defined as:

$$ICR(C_1, C_2) \triangleq \frac{1}{N_{1,2}} log(GLR(C_1, C_2)) \tag{2.11}$$

where

$$GLR(C_1, C_2) = \frac{Pr(\boldsymbol{O}_1|H_1)Pr(\boldsymbol{O}_2|H_1)}{Pr(\boldsymbol{O}_{1,2}|H_2)} \tag{2.12}$$

with H_1 and H_2 corresponding to the same hypothesis described for the ICSI system, $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$ corresponding respectively to the acoustic features in cluster C_1 and cluster C_2 and $\boldsymbol{O}_{1,2}$ to the union of the features of cluster C_1 and C_2. Mathematically, if each cluster C_1, C_2 and the union of the latter two C_1UC_2 are modelled by Gaussian Mixture Models with probability density functions f_1, f_2 and $f_{1,2}$, then equation (2.12) can be reformulated as:

$$GLR(C_1, C_2) = \frac{Pr(\boldsymbol{O}_1|f_1)Pr(\boldsymbol{O}_2|f_2)}{Pr(\boldsymbol{O}_{1,2}|f_{1,2})} \tag{2.13}$$

During the clustering process, speaker clusters with highest GLR score are merged, features realigned through Viterbi decoding and GMM speaker models re-estimated, until only one speaker cluster is left. All intermediate speaker segmentation and clustering hypothesis are stored for further processing. The best clustering hypothesis is then chosen according to a suitable quality metric.

2.6.2 Information bottleneck approaches

Other AHC diarization systems are based on the information bottleneck (IB) principle [33]. IB is a non-parametric framework that does not rely on any explicit modelling of the speaker clusters. Thus, the algorithm does not need to estimate continuously any GMM model for each of the speaker clusters resulting in lower computational complexity and faster than real-time systems while still reaching state-of-the-art diarization performance.

The IB principle is inspired by the Rate Distortion Theory in which a set of variables X is organized in a set of clusters C by minimizing the distortion between X and C. Given a set of variables Y relevant to the problem, for example in a document clustering problem a dictionary of words or in a speech recognition problem a set of relevant sounds, IB tries to find the right clustering C of the data X that conserve the highest information with the relevant variables Y. The clusters C can be interpreted as a compressed representation (bottleneck) of the entire data X, thus the information contained by X about Y is passed through the bottleneck C. The compression C should maintain as much as possible information with respect to the relevant variables Y, thus maximizing the mutual information $I(Y, C)$, and at the same time be the most compact coding of X, thus minimizing the mutual information $I(C, X)$. Mathematically, the optimum clustering $\hat{C}$ of the input data X is obtained according to the following maximization problem:

$$\hat{C} = \arg\max_{C} \left(I(Y, C) - \frac{1}{\beta} I(C, X) \right) \tag{2.14}$$

where β is the Lagrange multiplier representing the trade-off between the amount of information preserved $I(Y, C)$ and the compression of the initial representation $I(C, X)$.

In the context of speaker diarization, after feature extraction and SAD, the acoustic features are then clustered to obtain a uniform linear segmentation of the audio with each segment having an average segment length T_S. The input variables X are then defined as the initial speech segments output of the uniform segmentation. The relevance variables Y are chosen as components of a GMM model trained initially on the entire audio file. The Gaussian components of the estimated GMM models shares the same covariance matrix trained on the entire audio file. Once obtained the

initial segmentation of the audio and the relevance variables Y, the problem (2.14) can be solved by an agglomerative hierarchical clustering approach. The algorithm is initialized with an initial number of clusters corresponding to the initial segments. The segments are then iteratively clustered so that the decrease in the objective function is minimum. The operation stops when a unique cluster is reached. The optimal number of clusters is then selected according to a model selection criteria. After, the optimal number of clusters is selected, a Viterbi-based realignment is performed in order to refine the boundaries of the clusters that were obtained by the initial rough segmentation.

2.6.3 iVector based approaches

iVector based approaches are the most recent state-of-the-art methods for speaker diarization. Different off-line diarization systems based on the iVector speaker modelling approach reported in Section 2.14 have been proposed in recent years.

An initial work that explored the application of iVector modelling for two-speakers speaker diarization task is the one by Shum et al. [34]. Provided an audio stream, SAD is first applied to extract the relevant speech regions. Once the relevant regions of speech are extracted, they are further divided in speech segments of maximum duration T_S. From each of the obtained speech segments an iVector is then extracted. In order to retain only the most relevant speaker specific information Principal Component Analysis (PCA) is applied to the obtained iVectors in order to reduce their dimensionality. The reduced iVectors are then length-normalised and clustered through k-Means algorithm with a cosine-based distance metric. A resegmentation step is further applied in order to refine the initially rough segmentation boundaries using a Viterbi re-segmentation and Baum-Welch soft speaker clustering algorithm as explained in [13].

This work was then extended in [35] in the case of multi-speakers conversations. Spectral clustering algorithm [36] is applied in combination with k-Means clustering algorithm as an heuristic approach in order to determine the number of speakers in the conversation. Once the number of speakers is determined, PCA is applied to the speech iVectors to reduce the dimensionality and the reduced iVectors are then clustered always through k-Means clustering algorithm based on the cosine similarity distance. A resegmentation step is then applied in the same way as described above.

In [37] the authors present a complete off-line diarization system in which the clustering stage is based on a bayesian variational approach. The advantage of bayesian modelling approaches is in their natural preference versus simpler models to describe data and their resistance to over-fitting problems typical of maximum-likelihood approaches. A variational EM algorithms for Gaussian Mixture Models (VBEM-GMM) [38, 39] is used in order to cluster the initially extracted iVectors and to determine the final number of clusters.

In another work [40], the authors utilize a bottom-up approach with a PLDA-based similarity metric [41, 42]. The speech segments boundaries are determined through a BIC metric in order to determine the speaker change points. From the obtained speech segments, iVectors are then extracted. Speaker segments similarity is based on the PLDA metric. The merging of speakers clusters stop as soon as the similarity of all pairs of clusters is lower than a fixed threshold. The final number of clusters correspond to the final number of speakers. In [43] the authors propose a denser sampling of the audio in order to estimate the iVectors using longer speech segments. The audio is divided into speech segments of duration 1-2 seconds with 500 milliseconds of overlap with the preceding and following segments and iVectors are then extracted from each of the speech segments. Analogous to Prazak et al. [40] diarization system, the clustering is performed according to a PLDA-based similarity metric. The merging process is terminated as soon as the similarity between any pair of clusters does not exceed a threshold set analytically through a Bayesian decision process.

2.6.4 Binary feature vectors based approaches

Speaker modelling based on binary keys, described in subsection 2.4.5 represents a computationally efficient technique to model speaker clusters. By means of this modelling technique, the comparison and merging of speaker segments and clusters typical of bottom-up approaches reduces to the comparison of binary keys through binary metrics in favour of much more computationally efficient operations. The first bottom-up agglomerative diarization system based on binary keys was presented by Anguera et al. [44]. After a conventional initial segmentation of the speech regions, the initial speaker clusters are modelled through binary keys. Speakers clusters are iteratively merged by comparing the respective binary keys through simple similarity metrics. The process is repeated until only one speaker cluster is obtained. The best

clustering is thus chosen according to some information criteria. In following work in order to improve the system performance, Delgado et al. [45] have applied techniques such as Intra-Session and Intra-Speaker Variability (ISISV) compensation, explored alternative clustering selection methods and a faster way to train the KBM model by choosing the most discriminant Gaussian models. In another recent work, Delgado et al. [46] have also proposed to employ cumulative vectors which are compared according to a cosine distance rather than binary keys in order to preserve more information.

2.7 Divisive hierarchical clustering approaches

Although bottom-up approaches are the most common and diffused off-line diarization approaches, top-down or divisive hierarchical clustering approaches have revealed to reach comparable state-of-the art performance. In contrast with bottom-up approaches, top-down methods initially model all the speech segments of an audio stream with a single speaker model and add iteratively new speaker models until the correct number of speakers is reached and all speech segments are labelled. Only few top-down systems have been presented in literature [47, 48, 21]. Top-down approaches are more computationally efficient than bottom-up approaches and their performance can be improved through cluster purification [49]. The top-down system developed by LIA-EURECOM [21] together with the purification stage is utilised as an off-line baseline system in this dissertation and its state-of-the-art performance is set as an objective and goal for on-line diarization. The system is described thoroughly in Chapter 3, Section 3.4.

2.8 Integer Linear Programming based approaches

In [50, 51] the authors propose a new approach to replace the iterative bottom-up approach with a global process. The clustering process is formulated as a global Integer Linear Programming (ILP) and based on the iVector speaker modelling paradigm. The problem can be efficiently solved with any desired ILP solver. Each speech segment output of the SAD process is parametrized by an iVector for a total of N speech segments. The aim is to cluster the N iVectors into an optimal K number of clusters. The main assumption is that an iVector n can belong to cluster k if and only if the

distance between the center of the cluster (itself defined as an iVector) and the iVector is less than a set threshold. The goal consists of minimizing the number of clusters so that all the iVectors belong to only one cluster. Under this assumption, the problem can thus be formulated as a global ILP problem. The objective function z is to minimize the number of clusters K along with the dispersion of the iVectors within each cluster and is expressed as:

$$z = \sum_{k=1}^{N} y_k + \frac{1}{F} \sum_{k=1}^{N} \sum_{n=1}^{N} d(\boldsymbol{w}_k, \boldsymbol{w}_n) x_{k,n} \tag{2.15}$$

where:

- y_k is a binary variable indicating whether cluster k is selected,
- $x_{k,n}$ indicates whether iVector n belongs to cluster k,
- $d(\boldsymbol{w}_k, \boldsymbol{w}_n)$ is the distance between the center of cluster k and the iVector n,
- $\sum_{k=1}^{N} \sum_{n=1}^{N} d(\boldsymbol{w}_k, \boldsymbol{w}_n) x_{k,n}$ calculates the sum of the distances between the center of cluster k and the iVectors attached to that cluster,
- $\sum_{k=1}^{N} y_k$ calculates the number of clusters in the problem,
- F is a normalization factor to weights the subparts of equation (2.15).

As in reality the center of a cluster is itself an iVector, the distance between the center of a cluster k and the iVector n reduces itself to the computation of the a distance between the two iVectors. The speaker clustering problem can be thus rewritten as:

$$\begin{aligned}
&\text{minimize} && z \\
&\text{Subject to} && \sum_{n=1}^{N} x_{k,n} = 1, \forall k, && (2.16) \\
& && x_{k,n} - y_k \leq 0, \forall k, \forall n, && (2.17) \\
& && d(\boldsymbol{w}_k, \boldsymbol{w}_n) x_{k,n} \leq \delta, \forall k, \forall n, && (2.18) \\
& && x_{k,n} \in 0, 1, \forall k, \forall n, \\
& && y_k \in 0, 1, \forall k
\end{aligned}$$

Equation (2.16) ensures that all iVectors have been assigned to one cluster while equation (2.17) ensures that if iVector n is assigned to cluster k, then the cluster

k is selected. Finally, equation (2.18) guarantees that an iVector n can be selected from a cluster k if the distance is lower or equal to a fixed distance δ. This global clustering formulation has been also employed by Delgado et al. [46] with binary keys and cumulative vectors to improve the speed of a binary key cross-show speaker diarization system.

2.9 Hierarchical Dirichlet process hidden Markov model based approach

Non-parametric bayesian diarization solutions have been also proposed in literature by combining Hierarchical Dirichlet process (HDP) with HMM models. In 2006 Teh et al. [52] proposed initially the use of stochastic HDP to define a prior distribution on transition matrices over a countably infinite number of states of an HMM model. The resulting HDP-HMM based system is completely data-driven and the posterior distributions over the different states is inferred. Predictions of the number of states can be performed by averaging over different models of varying complexity. This work was then extended in [53] in order to allow a more robust learning of the speaker change dynamics and to provide an efficient and elegant way to estimate the number of participating speakers in a given audio stream.

2.10 On-line speaker diarization

All the speaker diarization systems presented in the previous sections are off-line and require the entire audio from the beginning till the end. However, with the increasing popularity of powerful, mobile smart devices, there is now a growing interest to develop on-line speaker diarization systems.

As illustrated in Figure 2.6, on-line speaker diarization analyses the audio available only up to a certain time τ to determine **"who is speaking now?"**. On-line diarization system determines which speaker is currently active, either a new one or an already encountered one, and its interval of activity. More formally, for each time τ, the system provides an optimised speaker segmentation $\tilde{G}_\tau$ and an optimised speaker sequence $\tilde{S}_\tau$ up to time τ.

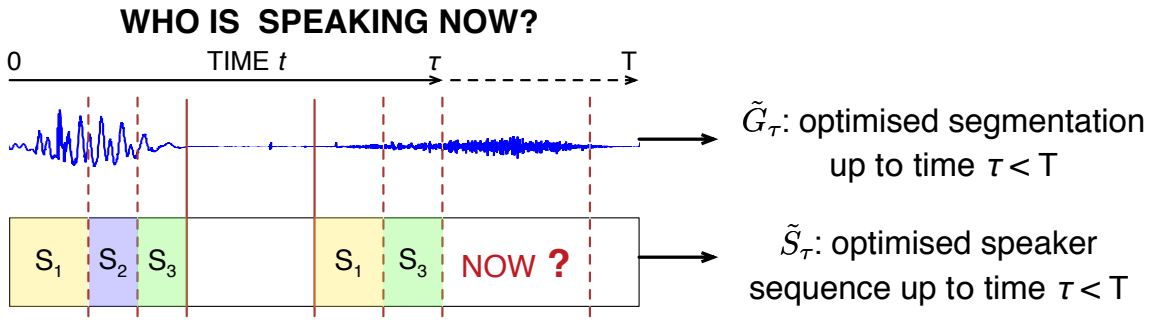


Fig. 2.6 An illustration of the on-line speaker diarization task.

Due to their computational complexity and high latency, the existing state-of-art diarization techniques described in previous sections are not easily adapted to on-line processing. Hence, there is an interest to develop entirely new, on-line approaches. On-line diarization still represents a challenging problem and its performance is pretty far from state-of-the-art off-line diarization systems.

The main reasons why on-line diarization represents such a challenging task might be synthesized in the following points:

- segmentation and clustering have to be performed in real-time,
- segmentation and clustering have to be performed by processing sequentially short speech segments in order to maintain a low latency with the risk of misclassification errors and bad updates of the speaker models,
- refining iteratively the segmentation boundaries and clustering on the past speech might be useful to estimate better models used to classify incoming speech segments, but with the drawback of increasing the latency of the system,
- on-line diarization is thus a trade-off between accuracy and system latency.

In contrast to the vast amount of research effort put in developing off-line diarization systems, only few works in literature have addressed the problem of on-line speaker diarization.

The problem of on-line diarization has been solved efficiently with the aid of multiple microphones and cameras. For example, in [54] the authors have developed a system that continuously captures the utterances and face poses of each speaker using a microphone array and an omni-directional camera positioned at the center of the meeting table. Through a series of advanced audio processing operations, an

overlapping speech signal is enhanced and the components are separated into individual speakers channels. Then the utterances are sequentially transcribed with the aid of a low-latency speech recognizer. In parallel with speech recognition, the activity of each participant (e.g., speaking, laughing, watching someone) and the circumstances of the meeting (e.g., topic, activeness, casualness) are detected and displayed on a browser together with the transcripts.

The problem of on-line diarization involving only two speakers has also been explored and even in this case efficient solutions have been proposed. For example, in [55] the authors present an on-line speaker diarization system for two-wire telephone conversations, thus involving only two speakers. Overlapping speech segments are parametrised by means of supervectors. A initial prefix of the audio stream is diarized off-line in order to train the initial speaker models, that are iteratively updated during the next on-line phase. The length of the prefix depends on the quality of the initially trained speaker models evaluated according to specific metrics. In a more recent work [56] an on-line speaker diarization based on iVectors is proposed. Speech segments are extracted according to the groundtruth and from each of them an iVector is extracted. Since most of the speaker information resides in the first dimensions of iVectors, the proposed system aims to estimate and update iteratively an on-line PCA transformation for the iVectors following a MAP based adaptation scheme. The transformed iVector at time n is then compared to all the previous iVectors up to time $n-1$ and classified according to the cosine metric.

Scenarios in which multiple speakers are involved and only a single microphone is available remain the most challenging ones. Moreover, the majority of on-line diarization systems recently proposed have focused on applications involving plenary speeches and broadcast news, where speaker turns are longer and there is less chance of overlapping speech. The first work worth mentioning is the one from Markov et al. [1]. In this work the authors describe a complete on-line speaker diarization system.

2.10.1 On-line segmentation and clustering

In Markov et al. [1], the authors present a complete on-line diarization system. The authors use a standard model based approach for the SAD stage. Non-speech events are represented by a single GMM model, while speech is modelled with two gender-dependent GMM models (male and female). For each acoustic feature, non-speech

and speech likelihoods are processed with two different median filters and according to their output features are assigned to being speech or non-speech. Start and end of a speech segment are then decided according to heuristic rules. Each speech segment is then classified according to the gender in a similar way as it is done for speech and non-speech.

Speaker modelling is based on GMM models. In order to decide if a speech segment i, parametrized by a set of acoustic features $\boldsymbol{O}_i$, is coming from a new speaker or one of the speakers already in the database, a testing hypothesis problem that results in a likelihood ratio is formulated as follows:

$$\boldsymbol{O}_i \in \begin{cases} S_{old}, & \text{if } LR(\boldsymbol{O}_i) > \theta \\ S_{new}, & \text{if } LR(\boldsymbol{O}_i) < \theta \end{cases} \tag{2.19}$$

where S_{old} is the class corresponding to the old speakers while S_{new} to the new speakers and θ a fixed pre-defined threshold.

The likelihood ratio $LR(\boldsymbol{O}_i)$ is defined as follow:

$$LR(\boldsymbol{O}_i) = \frac{Pr(\boldsymbol{O}_i|S_{old})}{Pr(\boldsymbol{O}_i|S_{new})} \tag{2.20}$$

where $Pr(\boldsymbol{O}_i|S_{old})$ is the highest likelihood of the speech segment against the old speaker models while $Pr(\boldsymbol{O}_i|S_{new})$ is the highest likelihood of the speech segment among the general male and female GMM models. In order to adapt and train the new speaker models the authors utilize an incremental variant of the EM algorithm.

In their next work Markov et al. [2], the likelihood ratio used to detect new speakers is refined in order to reduce the threshold variability due to the different number of speakers and to the gender by normalizing it in the following way:

$$LR^{norm}(\boldsymbol{O}_i) = \frac{LR(\boldsymbol{O}_i) - \mu_L}{\sigma_L} \tag{2.21}$$

where μ_L and σ_L are the mean and standard deviation of the likelihood ratios that can be initially estimated externally on a development dataset and later adapted on-line on the current audio.

Another work which has presented an on-line diarization system similar to Markov's one, above described, is the work Geiger et al. [3]. The main differences are the database

used and the usage of the MAP adaptation technique, introduced in subsection 2.4.2 rather than the incremental EM algorithm to update or introduce new speaker models. The authors develop an on-line diarization system for broadcast news databases. Initially, general speaker-independent GMM models for male and female speakers, music and non-speech related audio are trained on an external train dataset. Speech segments obtained after performing SAD are split into shorter segment according to a maximum duration T_S and are classified on-line.

For each segment i, parametrized by a set of acoustic features $\boldsymbol{O}_i$, the likelihoods against the GMM models in the repository are calculated. If the highest likelihood is obtained against the female or male GMM models then a new speaker model is obtained by MAP adaptation of the corresponding gender model with the acoustic features $\boldsymbol{O}_i$. The obtained speaker model is added to the repository of the GMM models and utilized to classify the next incoming speech segments, while the current speech segment i is labelled according to the new speaker model.

On the contrary, if the highest likelihood is obtained against one of the speaker models previously introduced, the corresponding speaker model is updated by MAP adaptation using the set of acoustic features $\boldsymbol{O}_i$ in the speech segment i. The segment is then labelled according to the detected speaker model. Finally, if the highest likelihood as been obtained against the music or non-speech models the segment is classified as music or non-speech and the corresponding model is always updated by MAP adaptation as in the other cases. At the end of the process, the audio will be labelled according to the number of speakers detected, music and non-speech.

Both, the previous systems are mainly based on a generic speech/non-speech segmentation where speech segments are classified according to generic male and female GMM models and speaker decisions are made at the end of each speech segment, output of the SAD process. In Oku et al. [57], the authors makes use of more sophisticated GMM models that model phonetic content of the audio. A low-latency, on-line speaker diarization system that exploits the phonetic information is developed in order to estimate more discriminative speaker models. Phone boundaries detected using a phone recognizer system are considered as potential speaker turns. Acoustic features are initially clustered into predefined acoustic classes and GMM models are trained with the same number of components as the number of acoustic classes. New speakers

are detected and clustered by using a delta-BIC distance to detect the speaker changes at each phonetic boundary.

The above described systems make use of the information only present in the incoming speech segments in order to perform classification and update the speaker models. However, if the incoming speech segments are of short duration, the information contained might not be enough to classify reliably the speech segments and to reliably update the speaker models. Thus, other work presented by Vaquero et al. [58] sought to utilize the state-of-the-art off-line diarization system proposed by ICSI [59] in order to continuosly refine the segmentation from the beginning of the audio up to a certain point and to update reliably speaker models used by the on-line classification of incoming speech segments.

An initial offline diarization stage is used to learn initial speaker models. As soon as initial speaker models are available, an on-line speaker identification system starts to classify the incoming speech segments of a certain fixed duration according to a maximum likelihood principle. The off-line diarization system runs always in parallel and in the background to diarize the audio available from the beginning up to a certain time T_i. The output labels are used to train new speaker models that will be then available to the on-line speaker identification system. The proposed system is illustrated in Figure 2.7. As it is possible to observe, performance is strictly dependent on the latency and accuracy of the off-line process in providing the labels to train new speaker models. Higher is the latency, higher will be the time needed by the on-line identification system to identify new speakers in the audio.

2.11 Summary

This chapter has presented an overview of the state-of-the-art in speaker diarization, which is mainly represented by off-line speaker diarization systems. These systems are characterised by high computational complexities and latencies. However, driven by the spread of smart objects, smart-phones and always listening sensors, on-line speaker diarization systems have received a growing interest. Despite this, the amount of proposed real-time diarization systems is minimal compared to off-line diarization systems. Moreover, the majority of on-line diarization systems has focused on the less challenging broadcast news or plenary speeches scenarios. These reasons motivate the

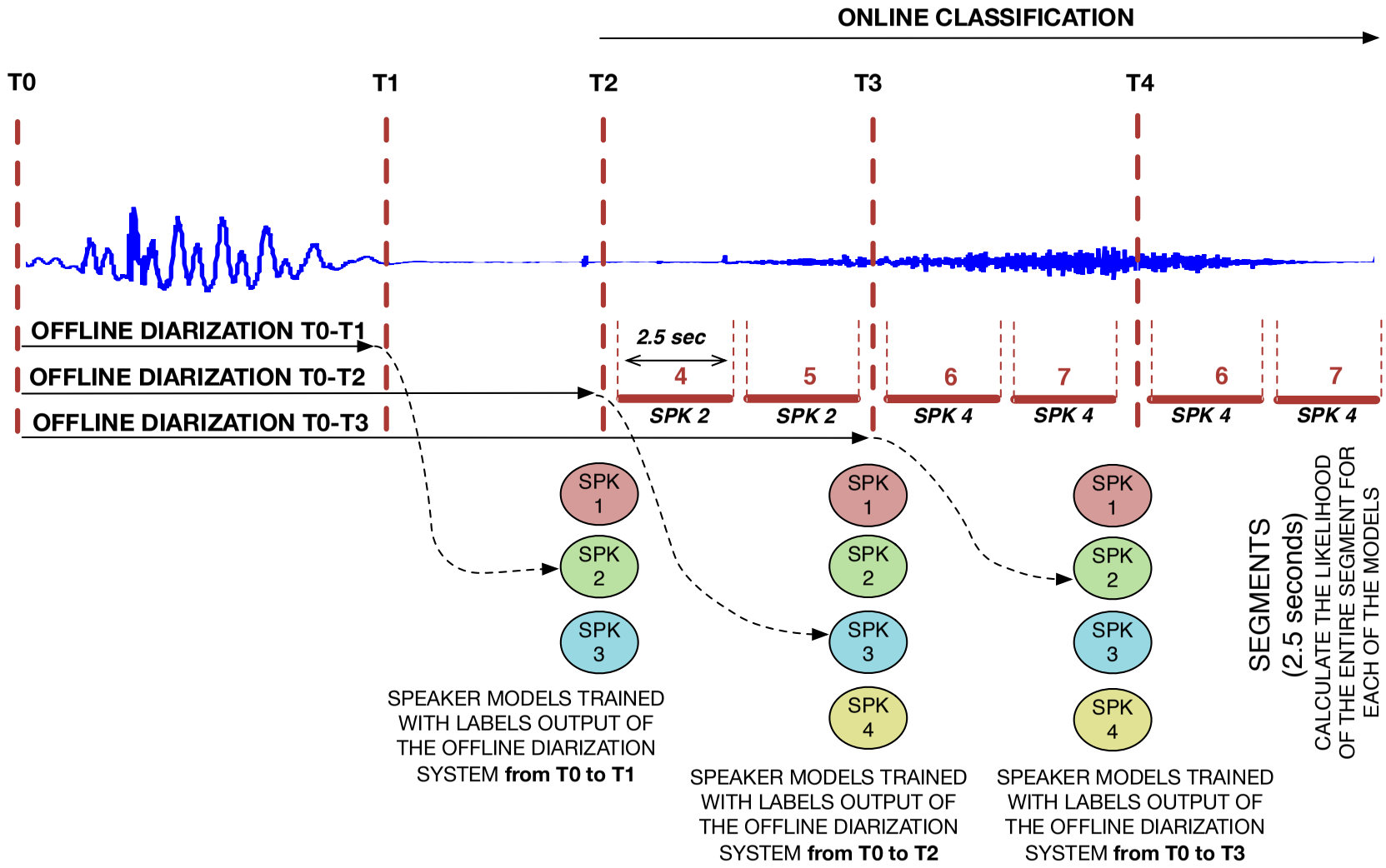


Fig. 2.7 An illustration of the hybrid diarization system presented by Vaquero et al. An off-line diarization system runs always in parallel and in the background to diarize the audio available from the beginning up to a certain time T_i. The output labels are used to train new speaker models that will be used by the online speaker identification system to classify the incoming speech segments.

aim of this dissertation in developing an efficient on-line speaker diarization system to support practical applications. Next chapter 3 will describe the main metric and datasets used for the development and evaluation of the proposed on-line speaker diarization systems presented in this dissertation.

Chapter 3

Metric and databases

In this chapter, the diarization error rate (DER) metric, the databases and the type of acoustic features used for the experimental work in this dissertation are described. As already mentioned, the majority of work in on-line diarization has been mainly addressing broadcast news scenarios and plenary speeches. These scenarios are characterised by longer speaker turns and lower spontaneity. Even though the development of an efficient on-line diarization system is mainly motivated by the increasing interest in the IoT and the spreading of speech-based context awareness applications, there are still no databases of recordings suited for these kind of emerging applications. At present, meeting recordings represent the most suitable data on which to develop an efficient on-line diarization system. The meeting recordings utilised for on-line diarization in this dissertation are amassed from the set of NIST RT corpora which were created for the NIST RT evaluations.

Finally, since this dissertation entails also the problem of phonetic variation, the TIMIT database used for the development and optimisation of PAT is also described.

The remainder of this Chapter is organised as follows: section 3.1 describes the diarization error rate metric used to measure the performance of on-line diarization. Section 3.2 briefly describes the NIST RT evaluations while section 3.3 provides an overview of the NIST RT corpora and the compiled datasets used for on-line diarization experiments. Off-line diarization baseline system is presented in section 3.4. Finally, the TIMIT dataset is described in section 3.5.

3.1 Diarization error rate (DER)

Diarization Error Rate (DER) is a metric to assess the performance of a speaker diarization system introduced by NIST for the RT evaluations. It is measured as the fraction of time that is not correctly attributed to a speaker or to non-speech. The standard diarization output contains an hypothesized sequence of speakers including the start and end time of each speech segment with a speaker label. The main purpose of speaker labels is to identify multiple interventions of the same speaker but without necessarily reflecting the real speaker identities. The quality of the output hypothesis is estimated by comparison to the ground-truth reference in order to obtain the overall Diarization Error Rate (DER). The DER score error can be decomposed into errors coming from three different sources:

- **Speaker error** (*SpkErr*): percentage of speech which has been assigned to the wrong speaker;
- **False alarm speech** (*FA*): percentage of speech present in the hypothesis but not in the ground-truth;
- **Missed speech** (*MS*): percentage of speech in the ground-truth which is not present in the hypothesis;

The DER is the sum of the above mentioned errors:

$$DER = SpkErr + MS + FA \tag{3.1}$$

More precisely and according to a standard dynamic programming algorithm defined by NIST, the DER score can be computed formally as:

$$DER = \frac{\sum_i \left(T_i^R \cdot \left(max\left(N_i^R, N_i^S \right) - N_i^C \right) \right)}{\sum_i \left(T_i^R \cdot N_i^R \right)} \tag{3.2}$$

where T_i^R is the duration of the i-th reference segment, and where N_i^R and N_i^S are respectively the number of speakers according to the reference and the number of speakers contained in the diarization hypothesis. N_i^C is the number of speakers that are correctly matched by the diarization system. As it can be seen from Equation (3.2), the DER is time-weighted meaning that it attributes less importance to speakers whose

overall speaking time is small. When evaluating performance, a collar around every reference speaker turn can be defined which accounts for inexactitudes in the labelling of the data. All the on-line diarization experimental results in this dissertation are computed by applying a collar of 25 milliseconds.

3.2 NIST RT evaluations

In the years 2004, 2005, 2006, 2007 and 2009, benchmark evaluations have been organised by NIST in the context of the Rich Transcriptions (RT) campaigns. The main aim of these evaluations was to facilitate the annotation and transcription of speech data by means of speaker diarization. The RT evaluations have revealed to be instrumental for the assessment of the state-of-the-art in speaker diarization by providing standard evaluation protocols, performance metrics and common annotated datasets. Although the initial RT evaluations addressed mainly broadcast news and telephone conversations scenarios, the latest RT evaluations have been focusing on meeting data characterised by a spontaneous speaking style. Each meeting was recorded with multiple microphones of different types and qualities usually positioned on the participants or spread around the meeting room. Depending on the arrangement of microphones into different classes, several evaluation conditions have been proposed by NIST. These include, the most common multiple distant microphones (MDM) and single distant microphones (SDM), individual headphone microphones (ICD) and all distant microphones (ADM). In the MDM condition, participants can utilise at the same time different recordings from different table-top microphones. In this case, beamforming could be applied in order to obtain a single pseudo channel or exploit inter-channel delay features in combination with conventional acoustic features to improve the diarization performance. Techniques for compensating the channel variation can also be applied. On the contrary, the SDM condition involves the use of only a single recording, usually from the central microphone. Therefore, beamforming, inter-channel delay features or ICD cannot be exploited.

In this dissertation, all the on-line diarization experiments have been carried out under the SDM condition since it is considered to be more challenging.

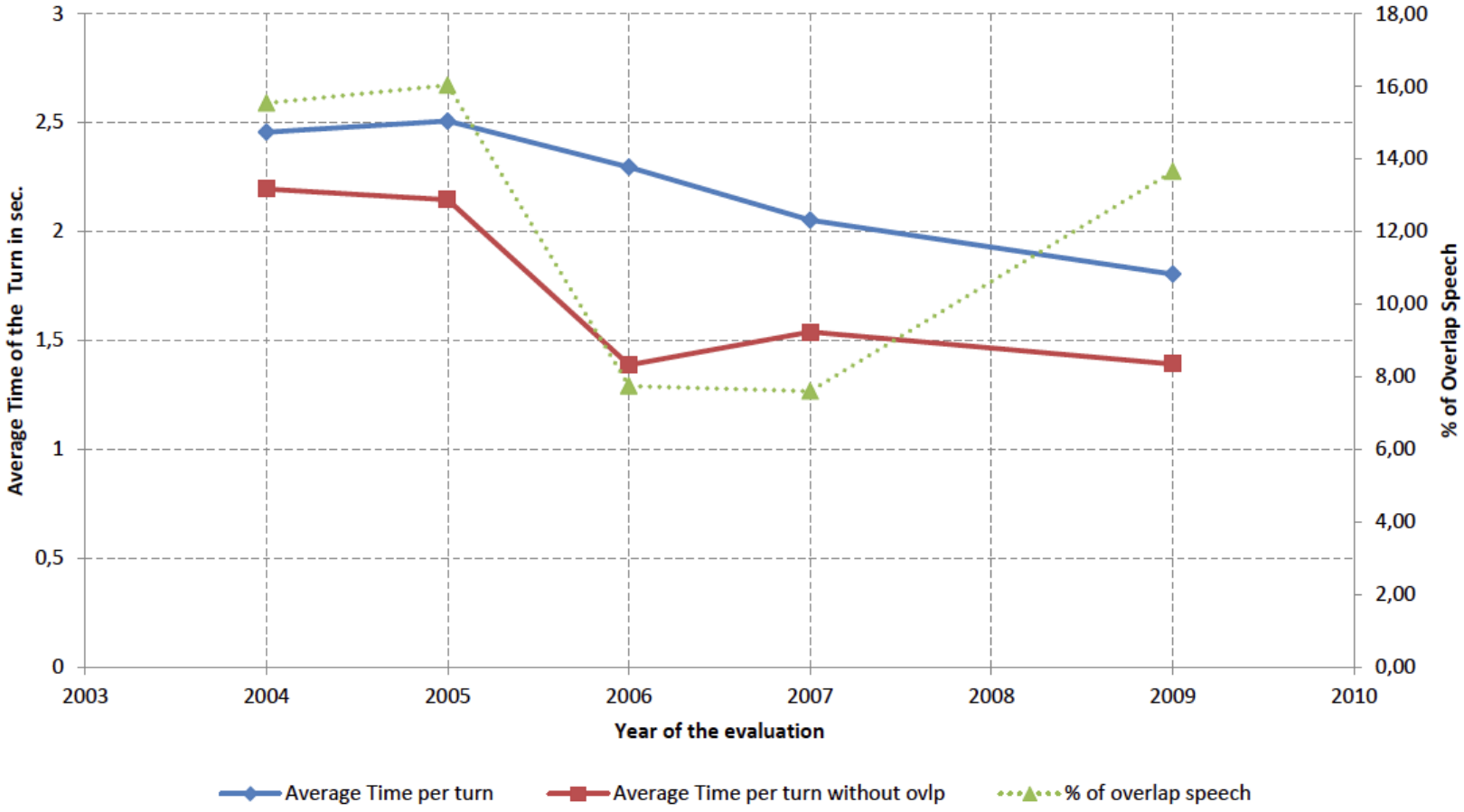


Fig. 3.1 Analysis of the percentage of overlap speech and the average duration of the turns for each of the 5 NIST RT evaluation datasets. Percentages of overlap speech are given over the total speech time (picture published with the kind permission of Simon Bozonnet).

3.3 RT meeting corpus

For each of the NIST RT evaluations, a new database of annotated audio meetings has been collected for a total of five meeting evaluation datasets. Figure 3.1 (taken from [60] and published in this dissertation with the author's permission) shows the difference among RT evaluation datasets in terms of speaker turn duration and overlap percentage. It is possible to observe that the last three evaluation datasets RT'06'RT'07 and RT'09 are characterised by shorter average turn durations, both when considering overlap and when not. This fact might suggest that meetings from the last three evaluations are more spontaneous and interactive and therefore more challenging for the speaker diarization task. From the figure it is also possible to observe that RT'04, RT'05 and RT'09 datasets are characterised by an average 15% of overlap speech, while RT'06 and RT'07 involve around 8% of overlap speech.

In order to carry out the experimental work on on-line diarization in this dissertation, three independent datasets have been compiled from the above mentioned NIST RT meeting corpus:

1. **RTubm:** a set of 16 meeting shows from the NIST RT'04 evaluations;
2. **RTdev:** a set of 15 meeting shows from the RT'05 and RT'06 evaluations, and
3. **RTeval:** a set of 15 meeting shows from the RT'07 and RT'09 evaluations.

The meeting IDs contained in the RTubm, RTdev and RTeval datasets are listed in Table 3.1. Clearly, there is no overlap between development and evaluation datasets even though they might contain recordings from the same place and possibly identical speakers. The average show duration within the RTubm, RTdev and RTeval datasets is 10, 15 and 24 minutes respectively while the average number of speakers within each set is 5, 5 and 4 respectively.

All the audio files from the RTubm, RTdev and RTeval datasets are pre-processed with Wiener filter moise reduction [61] in order to increase the signal-to-noise ratio. The obtained cleaned speech signals are then parametrised by means of mel-frequency cepstral features described in section 2.2 of Chapter 2. More precisely, for the unsupervised on-line speaker diarization experiments in Chapter 4, all audio files are characterised by 12 mel-frequency cepstral coefficients augmented by energy, delta and acceleration coefficients, thereby obtaining feature vectors with a total of 39 coefficients.

Instead, for the semi-supervised on-line speaker diarization experiments reported in Chapter 5, all audio files are characterised by 19 mel-frequency cepstral coefficients augmented by energy, thereby obtaining feature vectors for a total size of 20 coefficients.

The RTubm dataset is mainly meant for the training of general speech UBM models. The latter are generally trained by extracting the speech segments according to the ground-truth transcriptions and by 10 iterations of the EM algorithm. The RTdev and RTeval datasets are meant respectively for the development and evaluation of the on-line diarization systems proposed in Chapter 4 and 5.

In view of a better comparison with the other work in literature, on-line speaker diarization results are presented independently for the RT07 and the RT09 subsets both in Chapter 4 and Chapter 5. Finally, all the experiments to assess the performance of experimental on-line diarization systems are performed without considering speakers overlap which is removed according to ground-truth transcriptions.

3.4 Off-line diarization baseline system

The baseline off-line diarization system whose performance is set as reference for the development of an on-line diarization system is based on the official top-down system used for LIA-EURECOM's joint submission to the NIST RT'09 evaluations [21] and developed using the open source ALIZE toolkit. The purification stage proposed by Bozonnet in [49] has been introduced between the segmentation and re-segmentation stage while the final normalisation stage has been removed.

The final system is characterised by the following stages:

1. **Pre-processing**: all audio files are treated with Wiener-filter noise reduction [61]. Since in this dissertation only the SDM condition and not the MDM condition is addressed, neither beamforming techniques nor inter-delay channel features are utilised.

2. **Speech activity detection (SAD)**: the aim of this step is to separate speech segments from non-speech segments. This is performed using a two-states hidden Markov model (HMM), where each state is represented by a GMM with 32 Gaussian components, trained on external speech and non-speech data from the RT'04 and RT'05 evaluations. During this stage, the audio files are parametrised by 12 LFCC coefficients augmented by energy, delta and acceleration coefficients.

RTubm	
CMU_20020319-1400_d01_NONE	LDC_20011116-1400_d06_NONE
CMU_20020320-1500_d01_NONE	LDC_20011116-1500_d07_NONE
CMU_20030109-1530_d01_NONE	LDC_20011121-1700_d02_NONE
CMU_20030109-1600_d01_NONE	LDC_20011207-1800_d03_NONE
ICSI_20000807-1000_d05_NONE	NIST_20020214-1148_d01_NONE
ICSI_20010208-1430_d05_NONE	NIST_20020305-1007_d01_NONE
ICSI_20010322-1450_d05_NONE	NIST_20030623-1409_d03_NONE
ICSI_20011030-1030_d02_NONE	NIST_20030925-1517_d03_NONE

RTdev	
AMI_20041210-1052_d01_NONE	NIST_20050427-0939_d02_NONE
AMI_20050204-1206_d01_NONE	NIST_20051024-0930_d03_NONE
CMU_20050228-1615_d02_NONE	NIST_20051102-1323_d03_NONE
CMU_20050301-1415_d02_NONE	VT_20050304-1300_d01_NONE
CMU_20050912-0900_d02_NONE	VT_20050318-1430_d01_NONE
CMU_20050914-0900_d02_NONE	VT_20050623-1400_d02_NONE
ICSI_20010531-1030_d05_NONE	VT_20051027-1400_d02_NONE
ICSI_20011113-1100_d02_NONE	

RTeval	
RT07	**RT09**
CMU_20061115-1030_d01_NONE	EDI_20071128-1000_d01_NONE
CMU_20061115-1530_d01_NONE	EDI_20071128-1500_d01_NONE
EDI_20061113-1500_d01_NONE	IDI_20090128-1600_d01_NONE
EDI_20061114-1500_d01_NONE	IDI_20090129-1000_d01_NONE
NIST_20051104-1515_d03_NONE	NIST_20080201-1405_d03_NONE
NIST_20060216-1347_d03_NONE	NIST_20080307-0955_d03_NONE
VT_20050408-1500_d01_NONE	NIST_20080227-1501_d03_NONE
VT_20050425-1000_d01_NONE	

Table 3.1 Meeting IDs in the RTubm, RTdev and RTeval datasets.

Initially, Viterbi alignment is performed by using equal transition probabilities. Once the features are aligned, the speech and non-speech GMM models are MAP adapted with the corresponding features. These two steps are repeated for a maximum of 10 times until there are no more changes in the segmentation output. Heuristic rules might be applied to remove eventual rapid state transitions.

3. **Segmentation and clustering**: the speaker segmentation and clustering stage is based on an Evolutive Hidden Markov Model (E-HMM) where each state represents a particular speaker while the transitions represent speaker turns. All possible changes between speakers are authorized. The entire audio stream is first modelled with a single GMM speaker model and new speaker models are successively added to it until the full number of speakers are detected.All the speech segments from the audio are used to initialize an initial speaker model S_0 through the EM algorithm, representing the only initial state of the HMM. An iterative process is then started where a new speaker is added at each iteration. At the n^{th} iteration the longest speech segment is selected and used to train the n^{th} speaker model. A Viterbi realignment process is then started to find all the speech segments that fits to the new introduced speaker model. After realignment, speaker models are estimated again. This realignment/learning loop is repeated while a significant number of changes are observed in the speaker segmentation between two successive iterations. The current segmentation is analysed in order to decide whether the newly added speaker model n is relevant, according to some heuristic rules on the total duration assigned to speaker n. The stop criterion is reached if there are no more segments greater than 6 seconds in duration available with which to add a new speaker, otherwise the process starts to introduce another speaker. During this stage, the audio files are parametrised by 20 LFCC coefficients augmented by energy for a total of 21 coefficients.

   This segmentation and clustering stage is illustrated in Figure 3.2 for the case of two speakers conversation.

4. **Purification**: the purification stage has first been introduced by Bozonnet et al. [49]. It is performed after segmentation and clustering stage, in order to remove potential impurities in the obtained clusters which are supposed to contain a single speaker. First, a 16 component GMM is trained on the data of each

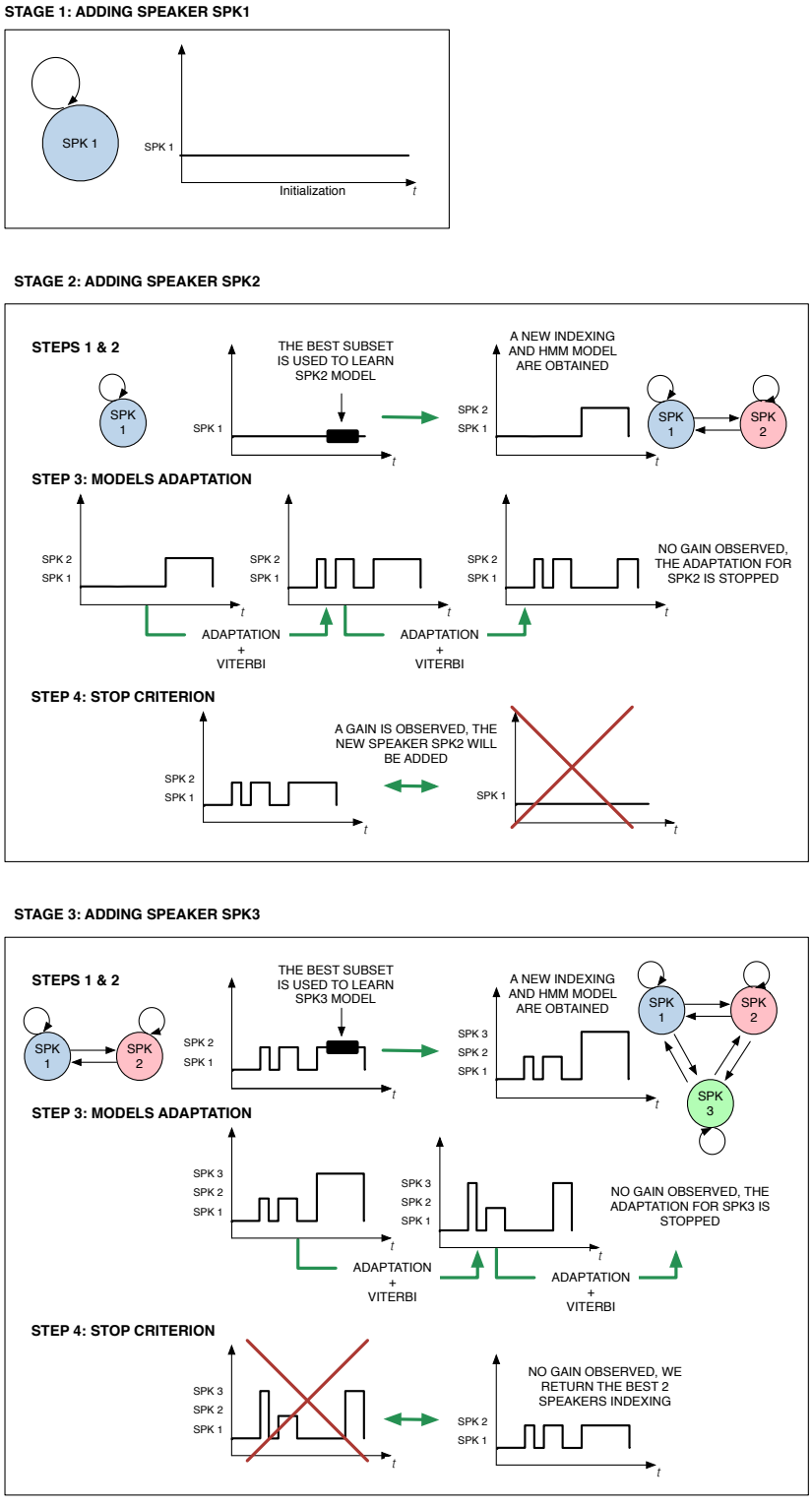


Fig. 3.2 Top-down speaker segmentation and clustering: case for two-speakers conversation.

speaker cluster by conventional EM algorithm. Each cluster is then split into sub-segments of 500 milliseconds duration and the top 55% segments which best fits the corresponding GMM are considered as classified. The remaining segments are then re- assigned to the closest GMMs by iterative Viterbi decoding and adaptation until all segments are classified. The acoustic features used for purification are the same as the ones used for the segmentation and clustering stage.

5. **Re-segmentation**: the re-segmentation stage aims to refine the speaker clusters by removing irrelevant speakers. During this stage, all speaker boundaries and segment labels are re-elaborated. An iterative training of a new HMM model from the segmentation output together with a Viterbi decoding are performed multiple times. Speaker models are adapted by MAP adaptation from a universal background model (UBM) trained on a Speaker Recognition corpus containing more than 400 speakers. Still, the acoustic features used for purification are the same as the ones used for the segmentation and clustering stage.

This off-line diarization system has already been optimised to attain the state-of-the-art performance. The type of acoustic features and complexity of the speaker models in each stage of such off-line systems differ when compared to the on-line diarization performance considered in this dissertation. The state-of-the-art performance of this off-line diarization system is regarded as the performance benchmark to which the proposed on-line systems in Chapters 4 and 5 aim to attain. However, it must be noted that the aforementioned benchmark is incomparable with respect to the developed on-line system due to the different segmentation and clustering techniques and especially the computational complexity.

The performance of the Top-Down diarization system obtained on the RTubm, RTdev and RTeval are reported in Table 3.2.

3.5 TIMIT dataset

The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus [62] was developed by a joint collaboration between Texas Instruments (TI) and Massachusetts Institute of Technology (MIT) in 1990 to advance acoustic-phonetic knowledge and to support research in automatic speech recognition (ASR).

Dataset	**DER** (%)
RTdev	17.02
RT07	18.24
RT09	19.20

Table 3.2 Performance of the baseline off-line diarization system for the RTdev and RTeval datasets.

Table 3.3 The setup of 38 phones used for PAT.

ENGLISH-LANGUAGE PHONES IN TIMIT ANNOTATIONS
hh, ih, z, eh, f, l, aa, b, ae, k, dh, dx, er, iy, m, n, g, r, ey, w, v, ah, y, uw, d, s, t, ng, p, sh, uh, ch, ay, ow, aw, th, jh, oy

In contrast with the NIST Rich Transcription datasets, characterized by the lack of accurate phonetic transcriptions, channel variation and different types of noises compromising the recordings, the TIMIT database is composed of high-quality, read English speech recorded with a close-talking microphone at 16kHz rate with 16 bit sample resolution. It was collected from a total of 630 speakers (192 female, 438 male) from 8 major dialect regions of the United States. All sentences were manually labelled and transcribed at the phonetic level.

TIMIT original phonetic transcriptions are based on 61 phones inspired by the alphabet ARPABET. For experimental purposes, these phones are usually collapsed into a set of 38 phones plus silence, as proposed by et al. in [63]. The set of final 38 phones are represented in Table 3.3.

Each speaker in the database contributes 10 short, phonetically-rich English language sentences whose average duration is 3 seconds for a total of 6300 sentences (5.4 hours). The prompts for the 6300 utterances consist of 2 dialect sentences (SA), 450 phonetically compact sentences (SX) and 1890 phonetically-diverse sentences (SI).

TIMIT database has represented a standard database for the speech community for many years and is still largely utilised nowadays, for both ASR and ASV experiments, due to its accurate phonetic labelling, its compactness, reduced level of noise and

channel variation, which permits benchmarking of systems' capabilities in a fast, reliable and controlled manner.

In this dissertation TIMIT is mainly exploited for the optimisation and development of PAT by means of ASV experiments described in Chapter 6. In this optic, the TIMIT database is divided in the following way:

- **TIMITubm**: 4620 speech recordings from a subset of 462 speakers of which 136 are female and 326 are male.
- **TIMITspk**: 9 speech recordings for each of the remaining 168 speakers of which 56 are female and 112 are male and for a total of 1512 speech recordings.
- **TIMITtest**: 1 speech recording for each of the remaining 168 speakers and for a total of 168 speech recordings.

Audio files are parametrised by 12 mel-scaled frequency cepstral coefficients (MFCCs) augmented by normalized energy, delta and acceleration coefficients thereby obtaining feature vectors with a total of 39 coefficients. All non-speech intervals from all the audio files of the TIMIT database are removed according to the ground-truth TIMIT transcriptions. The TIMITubm is used for the training of general speech UBM model by 10 iterations of the EM algorithm, whereas the TIMITspk dataset is meant for speaker models enrolment. The TIMITtest is instead utilised for ASV testing.

Due to the lack of proper datasets for diarization labelled at the phonetic level, in this dissertation TIMIT database is also exploited for the creation of simulated multi- speaker conversations. The obtained conversations are then used to analyse the potential benefits of the combination of PAT in combination with on-line speaker diarization in Chapter 7.

Chapter 4

Unsupervised on-line diarization

On-line diarization has attracted increasing interest in recent years due to the increasing popularity of powerful, mobile smart devices, the need for real-time information extraction in human interaction, growing interest in the Internet of Things (IoT) and the proliferation of always listening sensors.

Due to their computational complexity and high latency, the existing state-of-the-art off-line diarization techniques are not easily adapted to support on-line processing. Therefore, in recent years there has been an increasing interest to develop entirely new, on-line approaches.

The main contribution of this chapter entails the development of an unsupervised on-line diarization system for real-time applications. In contrast with the majority of works in literature which has focused on broadcast news and plenary speech scenarios, the proposed system is instead developed and optimised on meeting scenarios. Nowadays, meetings recordings represent the most challenging available data to develop an on-line diarization system for real-time applications.

The developed system is based on the sequential introduction and adaptation of speaker models by means of a sequential MAP adaptation algorithm. The performance of the system is assessed through experiments in which different segment durations T_S and different model sizes are used. Performance is also assessed in terms of dynamic convergence of the speaker models during time. Although in line with the performance of other on-line diarization systems presented in the literature which addressed less challenging datasets, the obtained error rates highlight the challenge involved in the development of an efficient on-line diarization system able to support practical applications.

The remainder of this Chapter is organized as follows. Section 1 outlines the sequential MAP adaptation algorithm used to initialize and adapt the speaker models. Section 2 describes the implemented unsupervised on-line diarization system. Section 3 describes the experimental setup and the obtained experimental results. Finally, a brief summary is provided in Section 5.

4.1 MAP adaptation

In contrast with off-line diarization, on-line speaker diarization necessarily requires the learning of speaker models iteratively, as soon as relevant data appears in the audio stream. Consequently, these are typically initialised using short speech segments. In particular, on-line diarization involves the comparison of similarly short, subsequent segments to the current inventory of speaker models and possibly their consequent re-adaptation using steadily amassed data.

MAP adaptation, initially introduced in Chapter 2, Section 2.4.2 is a model adaptation technique that allows the training of a GMM model when little amounts of training data are available. MAP adaptation is the starting point to develop an unsupervised on-line speaker diarization algorithm. Its main purpose is to introduce new speaker models as well as to update already introduced speaker models as soon as a speech segment is available. In this section, the conventional off-line MAP adaptation algorithm and the sequential MAP adaptation algorithm, both illustrated in Figure 4.1 and at the basis of the on-line diarization system, are explained.

4.1.1 Conventional maximum a-posteriori adaptation

The conventional MAP adaptation algorithm [27], the first algorithm illustrated in Figure 4.1, is commonly used to adapt a UBM model, generally trained with an EM algorithm, towards a specific speaker. The algorithm calculates the posterior probability of each Gaussian component given a set of training observations, and can be applied to update the mean, covariance and weight parameters of the Gaussian components which have the highest posterior probabilities. In the case where speaker specific training data is scarce, then the MAP adaptation of a well trained UBM generally gives better results than speaker specific models learnt directly by EM.

For a given speaker, let there be a sequence of D speech segments (D=4 in Figure 4.1) where each segment i is parametrised by a set of acoustic features $\boldsymbol{O}^{(i)} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_{M_i}$. As illustrated in Figure 4.1, conventional off-line MAP adaptation is performed using the UBM model λ_{UBM} and the pool of all D speaker segments. The sufficient statistics for the k-th Gaussian component are obtained as follows:

$$\begin{aligned} N_k &= \sum_{i=1}^{D} \sum_{m=1}^{M_i} Pr(k|\boldsymbol{o}_m, \lambda_{UBM}) \\ \boldsymbol{F}_k &= \sum_{i=1}^{D} \sum_{m=1}^{M_i} Pr(k|\boldsymbol{o}_m, \lambda_{UBM}) \boldsymbol{o}_m \\ \boldsymbol{S}_k &= \sum_{i=1}^{D} \sum_{m=1}^{M_i} Pr(k|\boldsymbol{o}_m, \lambda_{UBM}) \boldsymbol{o}_m^2 \end{aligned} \tag{4.1}$$

where $Pr(k|\boldsymbol{o}_m, \lambda_{UBM})$ represents the posterior probability of the k-th Gaussian component given the m-th observation $\boldsymbol{o}_m$. The MAP-adapted mean $\hat{\boldsymbol{\mu}}_k$, covariance $\hat{\boldsymbol{\sigma}}_k$ and weight $\hat{w}_k$ for the k-th Gaussian component are then given by:

$$\begin{aligned} \hat{w}_k &= \left(\alpha \frac{N_k}{\sum_{j=1}^{K} N_j} + (1-\alpha) w_k \right) \gamma \\ \hat{\boldsymbol{\mu}}_k &= \alpha \frac{\boldsymbol{F}_k}{N_k} + (1-\alpha) \boldsymbol{\mu}_k \\ \hat{\boldsymbol{\sigma}}_k &= \alpha \frac{\boldsymbol{S}_k}{N_k} + (1-\alpha)(\boldsymbol{\sigma}_k + \boldsymbol{\mu}_k^2) - \hat{\boldsymbol{\mu}}_k \end{aligned} \tag{4.2}$$

where γ is a normalization parameter such that $\sum_{k=1}^{K} \hat{w}_k = 1$ and where α is defined as:

$$\alpha = \frac{N_k}{N_k + \tau} \tag{4.3}$$

where τ is the pre-fixed scalar which regulates the relevance of the training data with respect to the UBM. The speaker model is then given by the tuple $s = (\hat{w}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k)$.

4.1.2 Sequential MAP

Sequential MAP is the second algorithm illustrated in Figure 4.1 and it is employed in order to introduce and update speaker models sequentially. Here, speaker models must

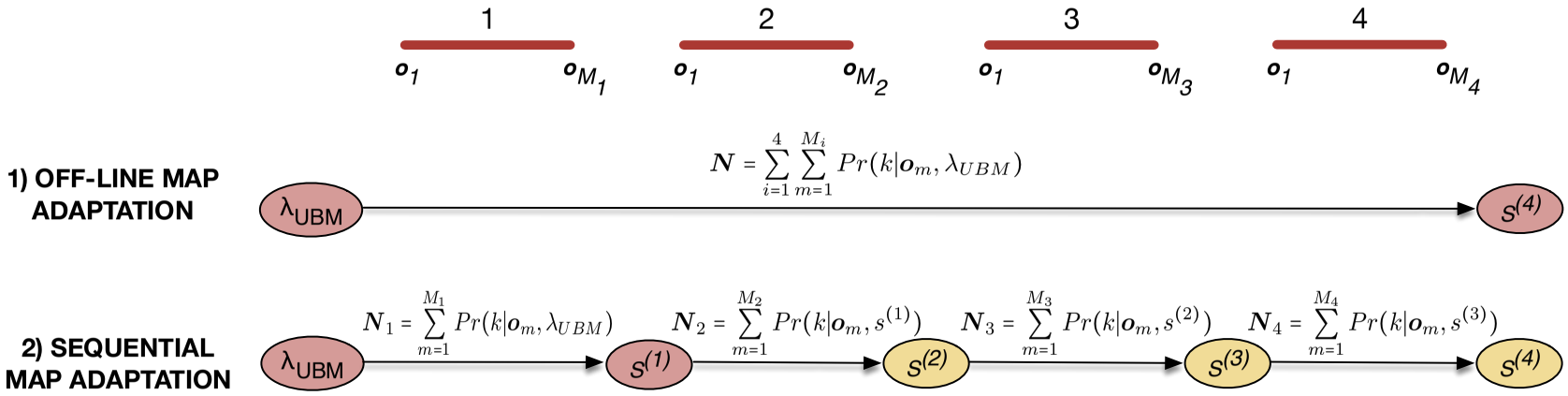


Fig. 4.1 A comparison of off-line MAP adaptation and sequential MAP adaptation for four speech segments from a particular speaker.

be updated continuously as and when new speech segments are assigned to any one of the speaker models in the current speaker inventory.

An initial speaker model $s^{(1)}$ can be trained by calculating the sufficient statistics of the first speaker segment using the same UBM model λ_{UBM}. The sufficient statistics calculated for the k-th Gaussian components are obtained from the application of (4.1) while setting $D = 1$. The mean, variance and weights of the updated model $s^{(1)}$ are similarly obtained from (4.2). As soon as a new speaker segment is available, then speaker model $s^{(i)}$ can be updated using the sufficient statistics of the speaker segment $i + 1$ and application of (4.1) with λ_{UBM} replaced by $s^{(i)}$:

$$
\begin{aligned}
N_{i+1} &= \sum_{m=1}^{M_{i+1}} Pr\left(k|\boldsymbol{o}_m, s^{(i)}\right) \\
\boldsymbol{F}_{i+1} &= \sum_{m=1}^{M_{i+1}} Pr\left(k|\boldsymbol{o}_m, s^{(i)}\right) \boldsymbol{o}_m \\
\boldsymbol{S}_{i+1} &= \sum_{m=1}^{M_{i+1}} Pr\left(k|\boldsymbol{o}_m, s^{(i)}\right) \boldsymbol{o}_m^2
\end{aligned}
\tag{4.4}
$$

where subscripts k have been omitted for simplicity. The mean, variance and weights of the updated model $s^{(i+1)}$ are then obtained in the usual way using (4.2).

The sequential MAP adaptation algorithm is at the basis of the new unsupervised on-line diarization system reported in the current chapter and whose implementation is described in the next section.

4.2 System implementation

The unsupervised on-line speaker diarization system developed for the diarization of meetings is illustrated in Figure 4.2. It is based on the baseline Top-Down or divisive hierarchical clustering approach to off-line diarization described in Chapter 3, Section 3.4 and the on-line diarization approach reported in [3]. Aside from background modelling, there are three stages: (i) feature extraction; (ii) speech activity detection and (iii) on-line classification.

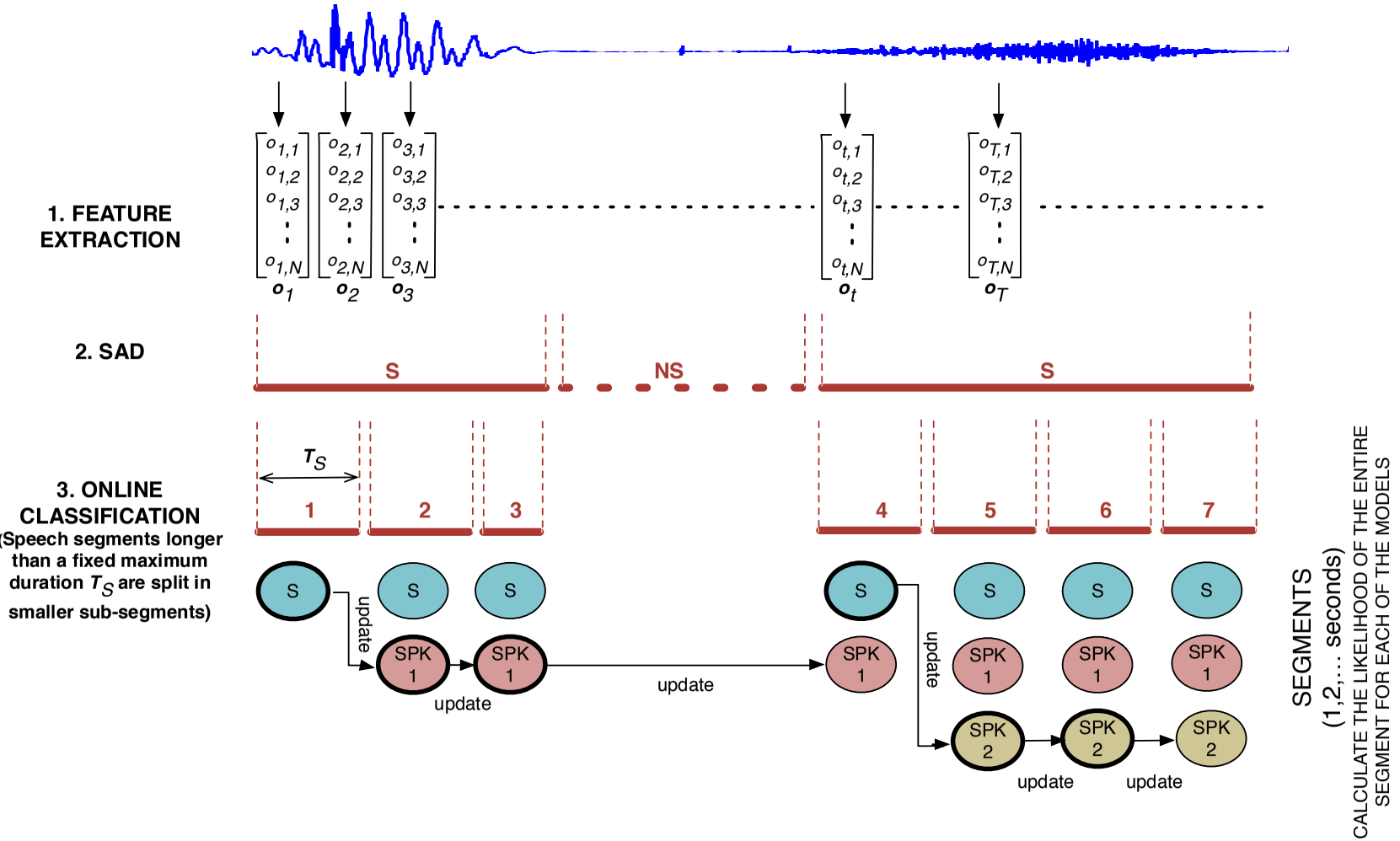


Fig. 4.2 An illustration of the on-line speaker diarization system.

4.2.1 Feature extraction and speech activity detection

The audio stream is first parametrised by a series of acoustic observations $\mathbf{o}_1, \ldots, \mathbf{o}_T$. Critically, for any time $\tau \in 1, \ldots, T$ only those observations for $t < \tau$ are used for diarization. Non-speech segments are removed according to the output of a conventional model-based speech activity detector (SAD) derived from the baseline Top-Down diarization system described in Chapter 3, Section 3.4. The remaining speech segments are then divided into smaller sub-segments whose duration is no longer than an a-priori fixed maximum duration T_S. Higher values of T_S imply a higher latency system. On-line classification is then applied in sequence to each segment.

4.2.2 On-line classification

Speech segments are either attributed to an existing speaker model, or a new speaker model is created. This procedure is controlled with a universal background model (UBM) denoted by s_0 which is trained on external data. New speaker models are introduced in the speaker inventory, if the current segment i generates a higher log-likelihood when compared to the UBM than to a set of speaker models s_j, where $j = 1, \ldots, N$ and where N indicates the number of speakers in the current hypothesis. Segments are attributed according to:

$$s_j = \underset{l \in (0,\ldots,N)}{\arg\max} \sum_{k=1}^{K} \mathcal{L}\left(\mathbf{o}_k | s_l\right) \tag{4.5}$$

where $\mathbf{o}_k$ is the k-th acoustic feature in the segment i, K represents the number of acoustic features in the i-th segment and where $\mathcal{L}\left(\mathbf{o}_k | s_l\right)$ denotes the log-likelihood of the k-th feature in segment i given the GMM model s_l. If the segment is attributed to s_0 then a new speaker model s_{N+1} is learned by MAP adaptation of the UBM model s_0 using the features contained in segment i. The segment i is then labelled according to the newly introduced speaker and N is increased by one. When a segment is attributed to an existing speaker, then the corresponding model is adapted through sequential MAP adaptation. The segment is then labelled according to the recognised speaker j as per Eq. 4.5.

4.3 Performance evaluation

The performance of the unsupervised on-line diarization system has been assessed by analysing the global DER as a function of the maximum segment duration T_S and the size of the speaker models and reported in Section 4.3.1. In addition to the global DER, dynamic convergence of the speakers models, represented by the evolving DER, is assessed as a function of time T_i and reported in Section 4.3.2. Finally, in Section 4.3.3 dynamic speaker statistics represent the average evolution of the number of speakers across different meetings.

4.3.1 Global DER

Diarization performance is first assessed globally in terms of the global average diarization error rate (DER) as a function of the segment duration T_S and for differing model sizes. Experiments have been performed for maximum segment durations of $0.25, 0.5, 1, \ldots, 10$ seconds and different UBM model sizes: 8, 16, 32, 64 and 128 Gaussian components. The UBM model s_0 is trained off-line by 10 iterations of the EM algorithm using the speech data from the RTubm dataset.

Left plots in Figures 4.3, 4.4 and 4.5 illustrate the on-line diarization performance in terms of global DER as a function of the segment duration T_S and model size for the RTdev, RT07 and RT09 datasets respectively. The optimal model size is either 32 or 64 Gaussian components, with the larger model size being the most consistent across the three datasets. In all cases, it is possible to observe that as the model size increases performance deteriorates further and probably due to the lack of sufficient data for reliable learning and adaptation of the speaker models. The optimal maximum segment duration T_S for all cases is around 3 or 4 seconds. Initially, the DER tends to decrease as the segment duration increases. As the segment size increases beyond the optimum, the global DER gets worse until it stabilizes. This is probably due to the fact that most of the speech segments after the SAD process are already shorter than the maximum segment duration T_S. Across the three datasets, the minimum DER is between 40% and 45%. This is a high error rate, but one not dissimilar to that reported in previous work performed using broadcast news data, e.g. [3]. The high diarization error rates might be caused by the initialisation of the speaker models through MAP adaptation of the speech UBM model with relative short speech segments. The initial

speaker models are not enough discriminative to reliably classify the incoming speech segments. While the application of adaptation and re-segmentation would improve the performance, they would also introduce further latency and computational complexity not in keeping with on-line diarization.

In the next chapter in order to overcome the bottleneck of speaker models initialisation with short speech segments, a semi-supervised system in which speaker models are seeded off-line with an initial amount of labelled training data is presented. Moreover, an incremental and more stable MAP adaptation technique to update the speaker models is introduced.

4.3.2 Adaptive speaker modelling and dynamic convergence

The average global DER errors calculated on each dataset as a function the segment duration T_S and different model sizes assess the overall performance of the system over all the entire audio. However, this measure does not perfectly reflect the way the classification accuracy evolves during the diarization process. In an on-line diarization system, it is interesting to analyse how the introduced and updated speaker models evolve in time in terms of classification accuracy. In this regard, dynamic convergence performance is assessed periodically at each minute T_i. Speaker models learned through the on-line diarization process only up until minute T_i are used to reclassify the speech segments of the entire audio according to the highest likelihood criteria, without any model adaptation or re-segmentation. The DER is then calculated on the output hypothesis at each minute T_i. The evolution of the DER as a function of time provides a measure of the evolution of the speaker models accuracy. While diarization performance should improve naturally as the full set of speakers is gradually introduced into the on-line process, 90% of speakers appear in the first 2 to 3 minutes of each show, as illustrated in Fig. 4.6 (red line). This approach to assessment is therefore still representative of on-line performance.

Right plots in Figures 4.3, 4.4 and 4.5 illustrate the dynamic convergence of the DER as a function of time T_i respectively for the RTdev, RT07 and RT09 datasets. Plots are again illustrated for segment durations T_S of between 1 and 6 seconds. All plots correspond to speaker models of 64 Gaussian components. In all cases the DER decreases towards and beyond the global DER as the amount of data available for model training increases. The plots also show that segment durations of less than 2

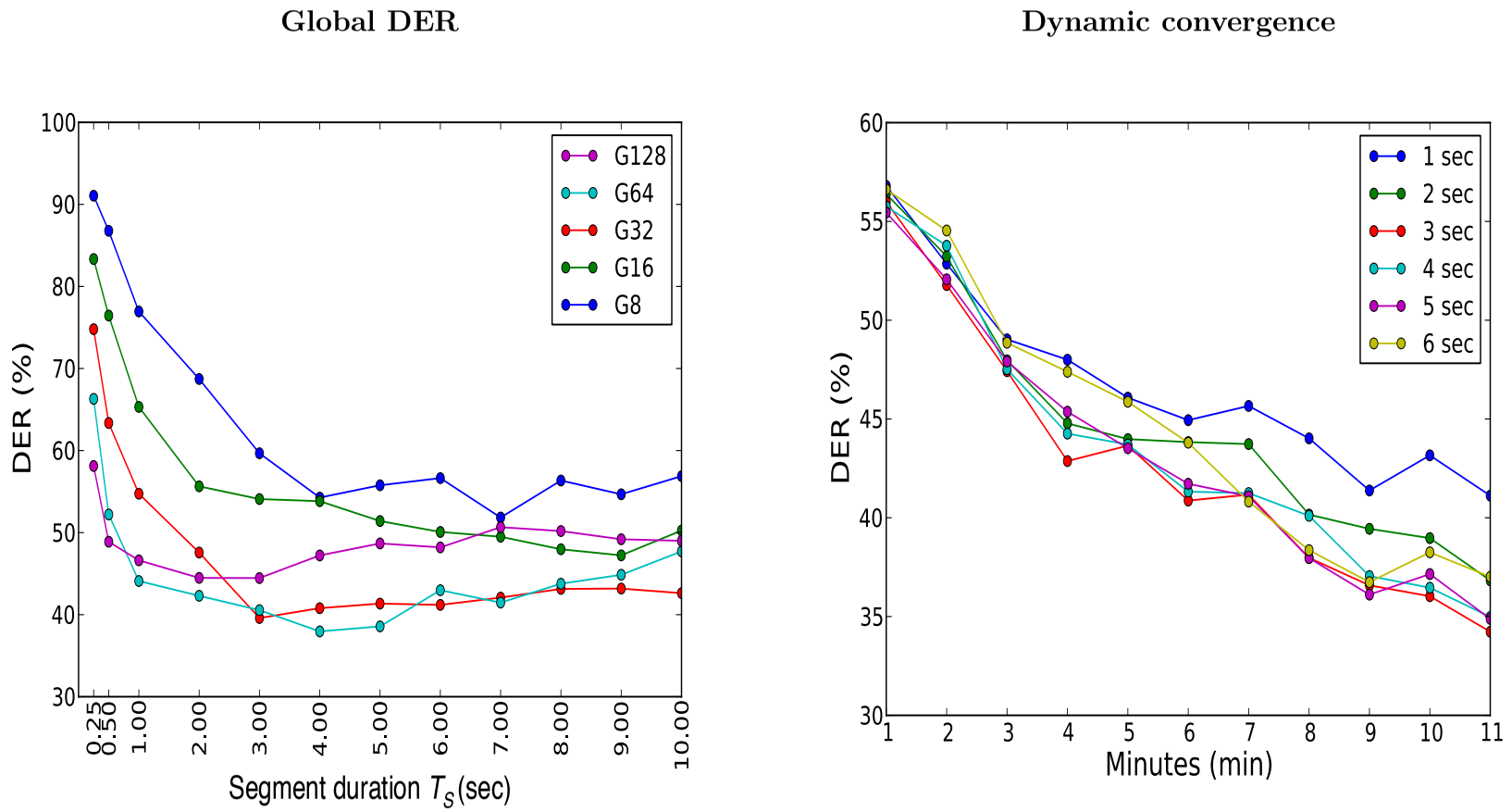


Fig. 4.3 Results are shown for the RTdev dataset. **Left plots:** an illustration of the global DER as a function of the segment duration T_S (0.25,0.5,1-10 sec) and for different model sizes (8-128).
Right plots: an illustration of the dynamic convergence of the DER as a function of time T_i.

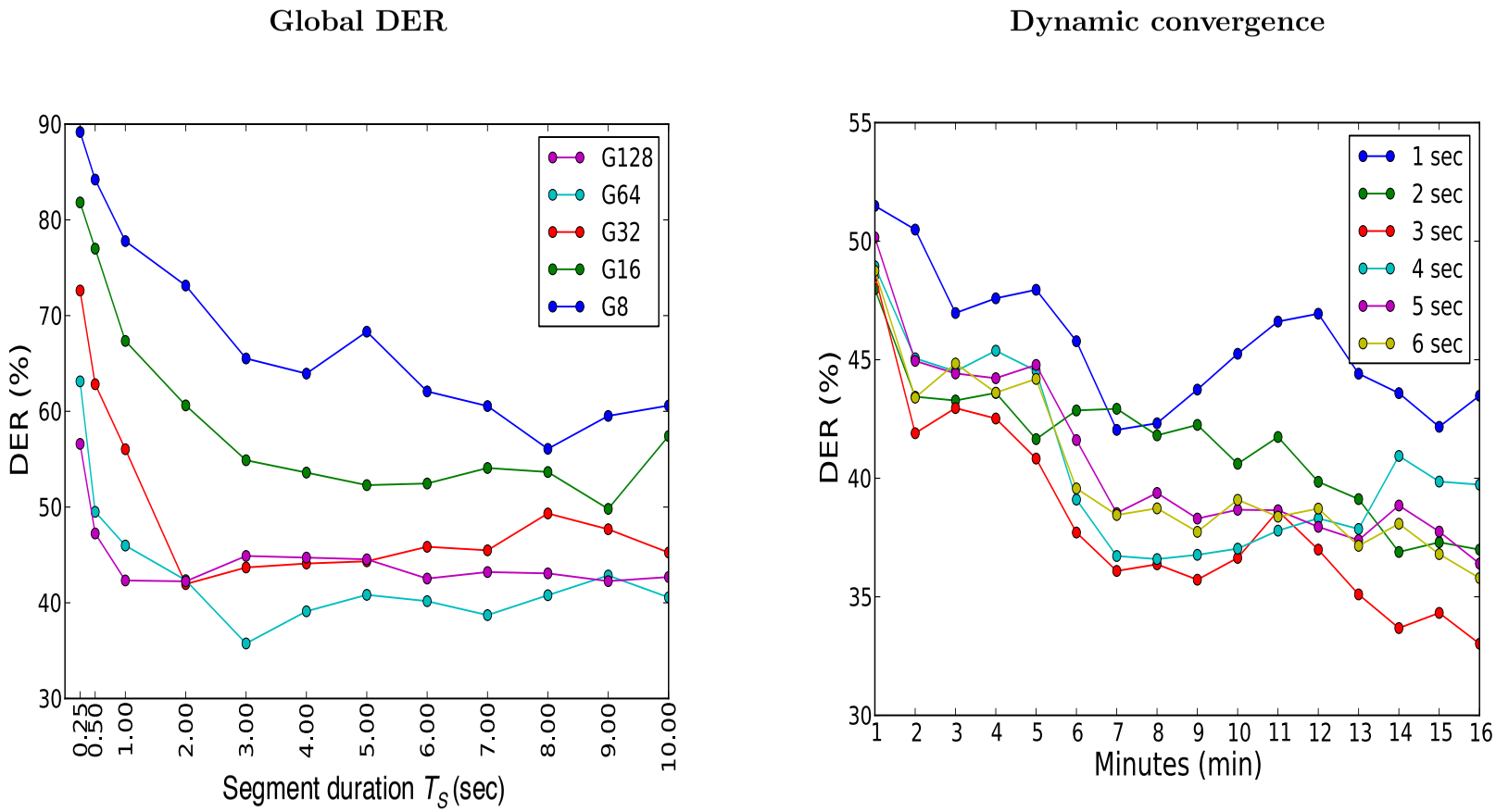


Fig. 4.4 Results are shown for the RT07 dataset. **Left plots:** an illustration of the global DER as a function of the segment duration T_S (0.25,0.5,1-10 sec) and for different model sizes (8-128).
Right plots: an illustration of the dynamic convergence of the DER as a function of time T_i.

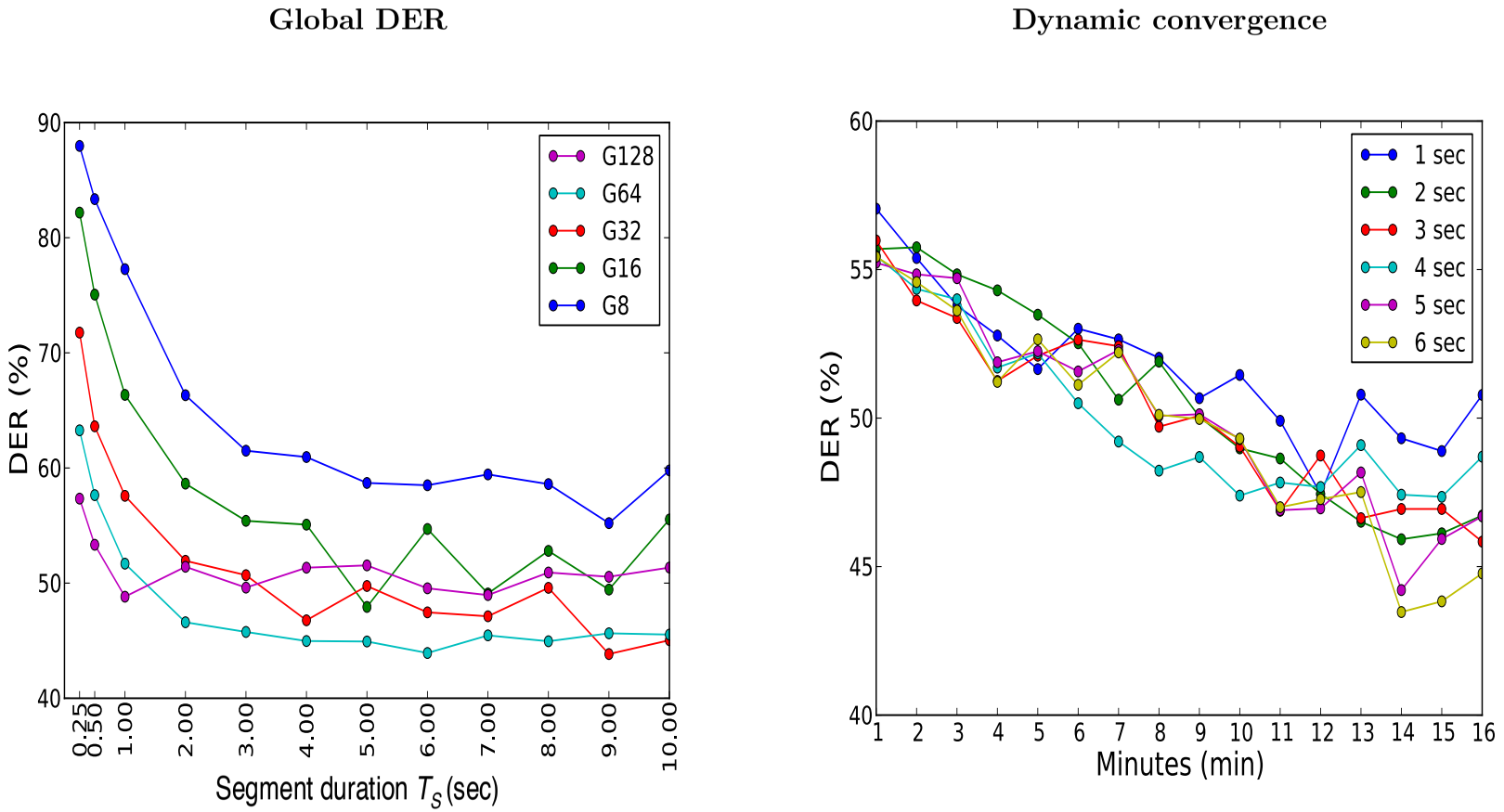


Fig. 4.5 Results are shown for the RT09 dataset. **Left plots:** an illustration of the global DER as a function of the segment duration T_S (0.25,0.5,1-10 sec) and for different model sizes (8-128).
Right plots: an illustration of the dynamic convergence of the DER as a function of time T_i.

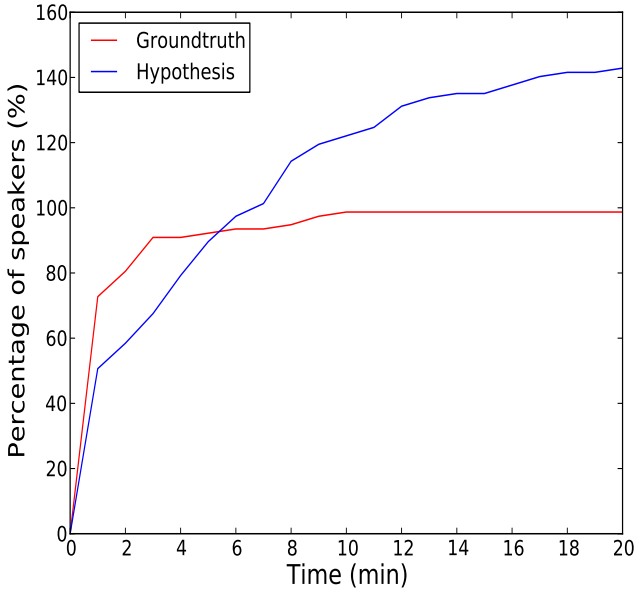


Fig. 4.6 An illustration of the evolution in speaker numbers for the RTdev dataset. Profiles shown for the ground-truth reference (red profile) and diarization hypothesis (blue profile).

seconds are largely insufficient for reliable diarization, with optimal performance being achieved with a segment duration T_S of 3 or 4 seconds. As the segment duration T_S increases, then more and more mid-segment speaker turns are missed, thus leading to higher DERs.

4.3.3 Dynamic speaker statistics

Figure 4.6 illustrates the evolution of the average speaker numbers for the RTdev set. Profiles are shown for the ground truth references (red line) and the corresponding number appearing in the automatically generated diarization hypothesis (blue line). The hypothesis corresponds to GMMs of 64 components and to a segment duration T_S of 2 seconds. For the first five to six minutes, the hypothesis contains fewer speakers than the ground truth whereas, beyond, the hypothesis contains more speakers than the ground truth. These observations suggest that an adaptive speaker penalty could be applied to favour the introduction of fewer speakers while reducing the rate at which new speakers are added later.

4.4 Summary

The majority of work in speaker diarization has evolved around the development of off-line diarization systems. However, driven by the popularity of powerful, mobile smart devices, the need for real-time information extraction in human interaction, growing interest in the Internet of Things (IoT) and the proliferation of always listening sensors, on-line diarization has attracted increasing interest in recent years. The work addressing on-line diarization and presented so far in literature have been focusing mainly on broadcast news plenary speech scenarios rather than the more challenging meeting scenarios characterised by higher spontaneity and shorter speaker turns.

This chapter has presented our efforts to develop a new adaptive, unsupervised on-line approach to speaker diarization based on a sequential MAP adaptation approach for the more challenging meeting data captured with a single distant microphone. The performed experiments have shown that the best performance implies a latency in the order of 3 or 4 seconds and the accuracy of the trained speaker models converges as the amount of training data increases. While results are in line with those reported for less challenging data, diarization error rates remain high, probably too high to support any practical applications.

In the next chapter by means of small scale ASV experiments it is shown that the main bottleneck of on-line diarization lies unsurprisingly in the amount of training data used to initialise the speaker models. To overcome this bottleneck, a semi-supervised on-line diarization solution in which speaker models are initially seeded with different amounts of training data and updated continuously by means of a newly introduced incremental MAP adaptation technique, is presented.

Chapter 5

Semi-supervised on-line diarization

Chapter 4 entails our first attempt to develop an unsupervised on-line diarization system. Unfortunately, the obtained diarization error rates on the most challenging meeting datasets are high, probably too high to support any practical applications. There is thus a strong need to investigate alternative strategies. In this chapter it is initially shown unsurprisingly that the main bottleneck in the development of an efficient on-line diarization system lies in the quantity of data used for speaker model initialisation. Two possible solutions to mitigate this bottleneck involve either the relaxation of latency(on-line) or supervision constraints. Since the former is at odds with the pursuit of an on-line diarization system, this chapter investigates various semi-supervised approaches.

While semi-supervised approaches have been reported previously for off-line diarization [64], the first contribution of this chapter is the development of a new, semi-supervised on-line diarization system. Based upon the approach described in Chapter 4 and with the number of speakers assumed to be known a-priori, the new system exploits short amounts of labelled speech for supervised speaker model initialisation. The remainder of the process remains entirely unsupervised.

While knowing the number of speakers and the use of labelled data is also at odds with the traditional definition of diarization, many practical scenarios, for instance family members interacting with different smart objects which deliver personalised services or workers in the same meeting room equipped with a system able to display specific information in real-time depending on who is currently speaking, allow a short-duration phase during which each speaker introduces themselves. This data may be used readily for initialisation. Despite the manual labelling of such intervals being

an inconvenience, it is perhaps a price worth paying for the significant improvement in diarization performance.

The main goal of the work presented in this chapter is thus to determine what duration of manually labelled speech is required in order to deliver satisfactory performance. This is defined as the achievable state-of-the-art off-line diarization. The second contribution of this chapter relates instead to an incremental approach to on-line model adaptation, which proves instrumental in delivering low diarization error rates.

The remainder of this chapter is organized as follows. Section 2 demonstrates the challenge faced in on-line diarization and justifies the need for relaxed supervision constraints. Section 3 describes the incremental model adaptation procedure and the new semi-supervised, on-line diarization system. Section 4 describes experimental work whereas a brief summary is provided in Section 5.

5.1 Speaker modelling

In an on-line diarization scenario, speaker models are typically initialised using short speech segments. Diarization involves the comparison of similarly short, subsequent segments to the current inventory of speaker models $\tilde{\Delta}$ and possibly their consequent re-adaptation using steadily amassed data. While necessary to meet the requirements for on-line processing, the use of short segments for both operations also ensures inter-segment speaker homogeneity.

These two operations form the essential elements of *speaker verification*, namely speaker enrolment and testing. It is well known that the reliability of both depends fundamentally on data duration. The work presented in this section aims to examine the dependence of *speaker diarization* on segment duration and hence to illustrate the potential to improve on-line diarization performance. This examination is performed through strictly controlled automatic speaker verification (ASV) experiments which avoid complications associated with overlapping speakers and compounding diarization nuances.

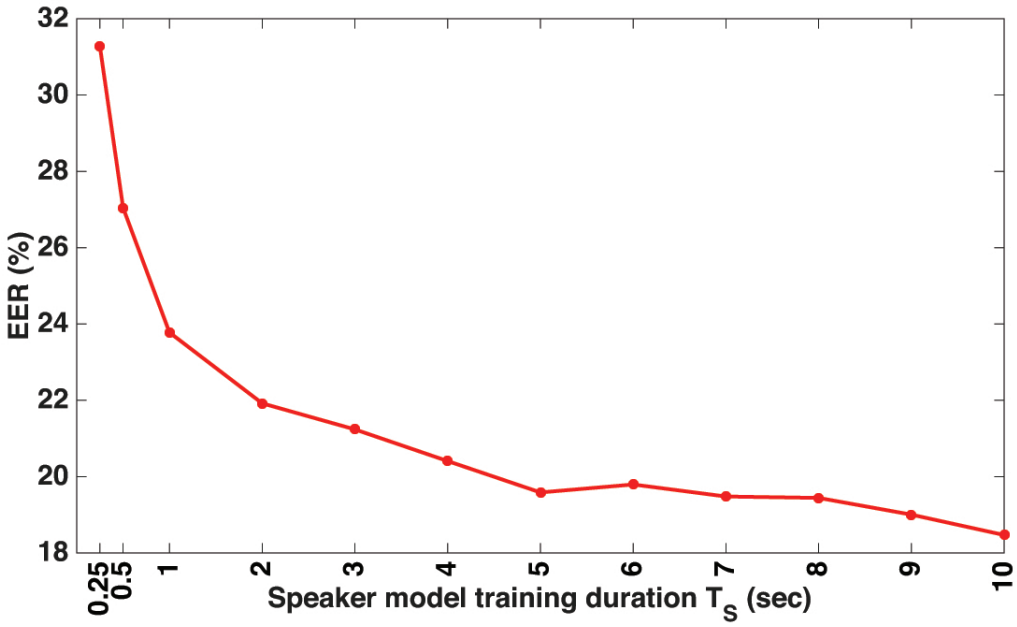


Fig. 5.1 EER as a function of T_S, namely the quantity of data used to train the speaker models .

5.1.1 ASV experiments

The system used for ASV experiments is a conventional GMM model with universal background model (GMM-UBM) system. The UBM of 64 Gaussian components is trained on the RTubm dataset with 10 iterations of expectation-maximisation (EM).

The speech data of all speakers in the RTdev and RTeval datasets with a floor time greater than 20 seconds are identified using the ground-truth references. The data for all other speakers are discarded. The first 10 seconds of speech of each speaker are set apart for model training while the remaining speech segments are used for testing. All speech segments are further divided into sub-segments of maximum duration T_S where $T_S = 0.25, 0.5, \ldots, 10$ seconds.

Training data, of identical duration T_S is randomly selected from the 10-second training segment and speaker models are derived from the UBM using MAP adaptation with a relevance factor set to 10, whereas testing is performed separately on every single, same-length sub-segment in the test data. Exhaustive testing is performed for all speakers; all test segments are compared to all speaker models. This equates to a large number of short-duration target and impostor trials from which ASV performance can be gauged in the usual way.

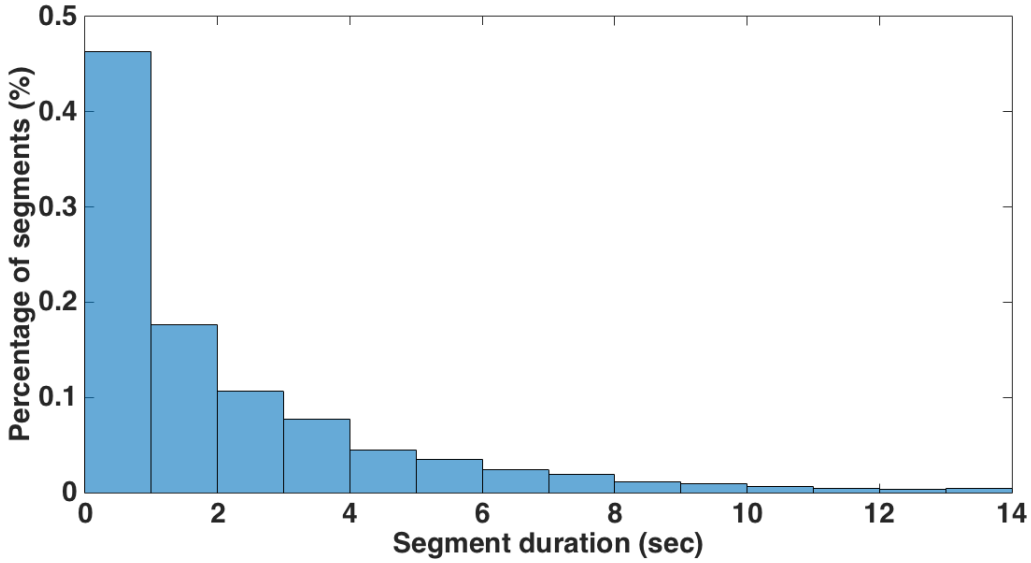


Fig. 5.2 Speech segment duration distribution for the RTdev dataset.

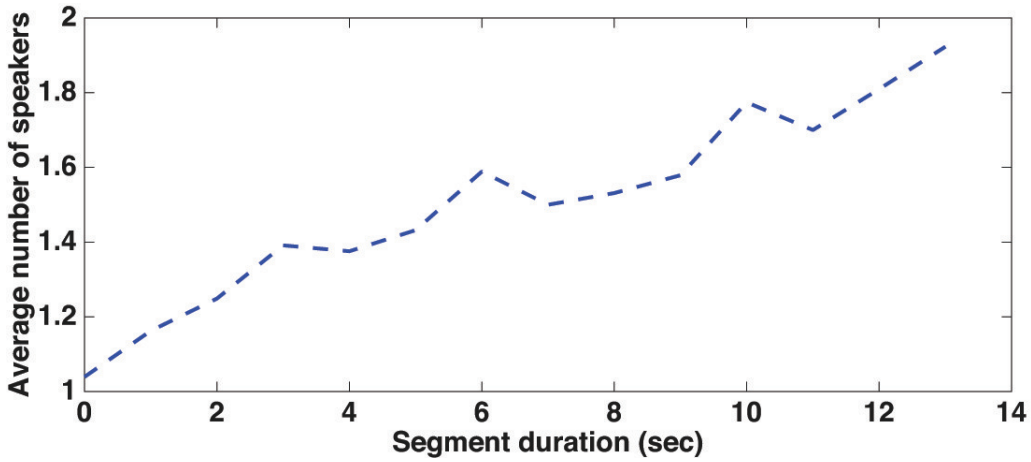


Fig. 5.3 Average number of speakers as a function of the speech segment duration for the RTdev dataset.

ASV results, combined for RTdev and RTeval, are illustrated in Figure 5.1 in terms of the equal error rate (EER) as a function of T_S. Unsurprisingly, performance improves as T_S increases. Critically, with very low quantities of training and testing data less than 1 second in duration, the EER is extremely high. Lower EERs are observed for data quantities of 10 seconds. The elbow is around 5 seconds, where the EER is in the order of 20%. Even with a value of $T_S = 10$ seconds, the EER is perhaps still high for what is essentially same-session ASV. This is probably due to the fact that most speech segments are considerably shorter than the value of T_S.

Figure 5.2 illustrates the segment duration distribution for the RTdev dataset. The vast majority of segments are seen to be less than 5 seconds in duration. The use of longer segments in on-line speaker diarization applications also comes at the increased risk of speaker model impurities.

Figure 5.3 illustrates the average number of speakers as a function of segment length. The plot shows that, beyond segment lengths of 5 seconds, a segment is more likely to contain 2 speakers than 1 speaker. Added to this, the use of longer segments would entail greater latency, which is at odds with the need for on-line diarization.

While admittedly trivial, this analysis shows that, in independence from overlapping speech and diarization nuances, the potential for successful on-line diarization is severely limited by the potential to acquire sufficient, speaker-homogeneous training and testing data. In summary, reliable decisions cannot be made when models are initialised on such short segments of speech. These observations call for an alternative approach to on-line diarization.

5.2 Semi-supervised on-line diarization

In previous section it has been shown by means of ASV experiments that the main bottleneck in on-line speaker diarization relies in the use of short-duration speech segments for speaker model initialisation. Although short-duration speech segments provide a lower system latency, the derived speaker models are not sufficiently reliable for proper classification. On the contrary, the use of longer duration speech segments, even though would provide a larger amount of training data, would increase the system latency and the risk of including impurities, i.e. more speakers in the same speech segment. An alternative solution to overcome this problem takes the form of supervision constraints. Speaker models of the participants involved are seeded with an initial amount of labelled training data. By following this direction, the open question which needs to be answered is then: **what quantity of seed data is required to reach the same diarization performance obtained with an off-line system?** Before describing the implemented semi-supervised on-line diarization system, the new incremental MAP adaptation procedure for the updating of speaker models is described.

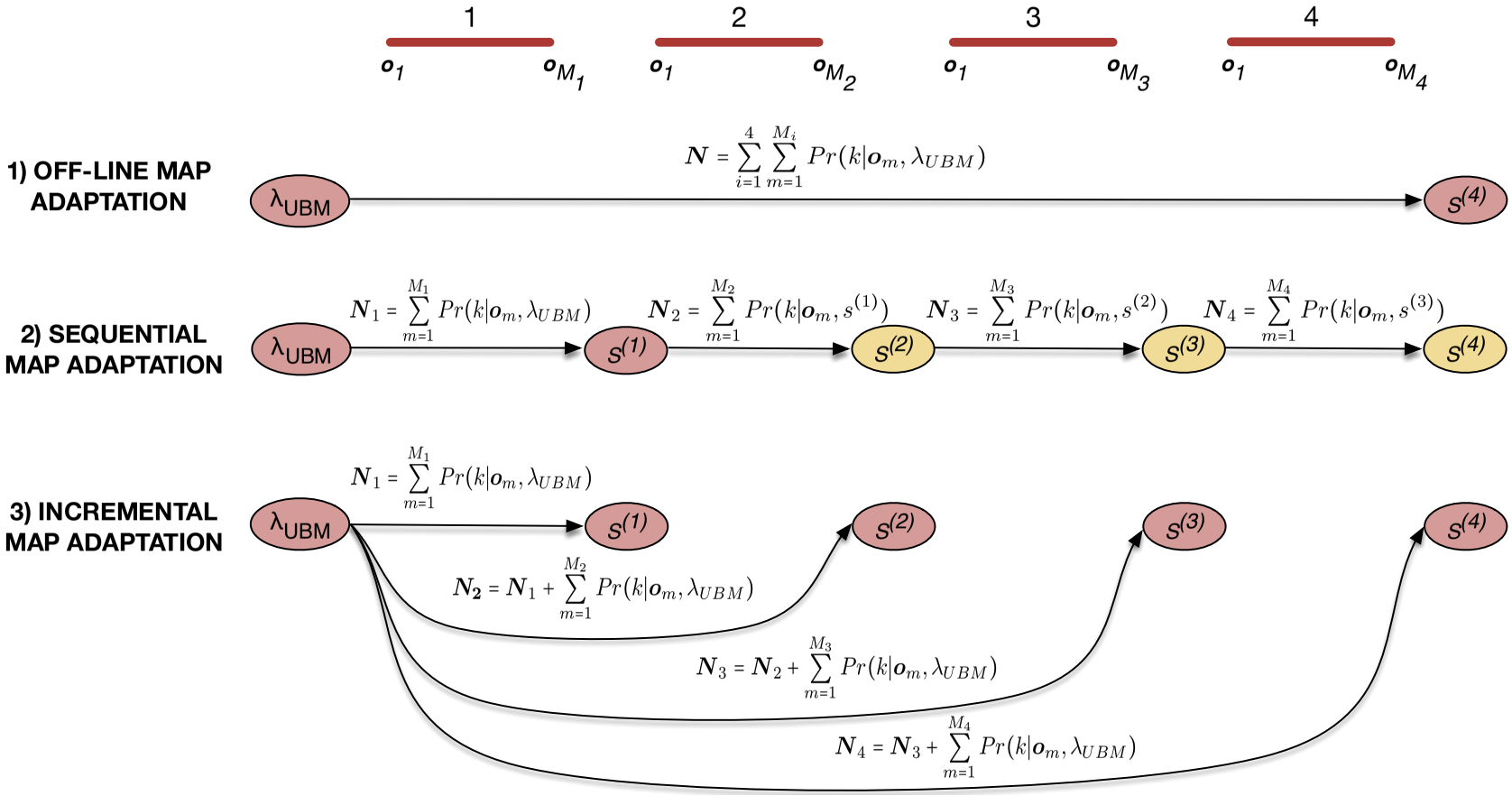


Fig. 5.4 A comparison of off-line MAP adaptation, sequential MAP adaptation and incremental MAP adaptation for four speech segments from a particular speaker.

5.2.1 Incremental MAP

The initialisation and update of speaker models in the unsupervised on-line diarization system described in Chapter 4 is guaranteed by a sequential MAP adaptation procedure, in which the speaker models are sequentially MAP adapted as soon as a speech segment is available. However, in a semi-supervised scenario the seeding of the speaker models with labelled training data allows the update of the speaker models by means of a more robust incremental MAP adaptation procedure.

For a given speaker, let there be a sequence of D speech segments (D=4 in Figure 5.4) where each segment i is parametrised by a set of acoustic features $\boldsymbol{O}^{(i)} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_{M_i}$. As explained in Chapter 4, Section 4.1.2 in the sequential MAP adaptation procedure, the second algorithm illustrated in 5.4, the sufficient statistics $N_{i+1}, \boldsymbol{F}_{i+1}$ and $\boldsymbol{S}_{i+1}$ for the speaker segments $i+1$ are calculated against the previous model $s^{(i)}$ and depend non-linearly on $N_i, \boldsymbol{F_i}$ and $\boldsymbol{S_i}$ in terms of Gaussian occupation probabilities. Accordingly, even given the same observations in the same segments, the speaker models obtained from the conventional, off-line and sequential MAP adaptation procedures are not the same. However, in the newly proposed incremental MAP adaptation approach, the third algorithm illustrated in Figure 5.4 the sufficient statistics are calculated always with respect to the general speech UBM model λ_{UBM} and accumulated during time.

Here, the initial speaker model $s^{(1)}$ is obtained in the same way as with sequential MAP adaptation. In order to update the speaker model $s^{(i)}$, sufficient statistics for speaker segment $i+1$ are now always calculated with the original λ_{UBM} model and accumulated with sufficient statistics N_i, $\boldsymbol{F}_i$ and $\boldsymbol{S}_i$:

$$
\begin{aligned}
N_{i+1} &= N_i + \sum_{m=1}^{M_{i+1}} Pr\left(k|\boldsymbol{o}_m, \lambda_{UBM}\right) \\
\boldsymbol{F}_{i+1} &= \boldsymbol{F}_i + \sum_{m=1}^{M_{i+1}} Pr\left(k|\boldsymbol{o}_m, \lambda_{UBM}\right) \boldsymbol{o}_m \\
\boldsymbol{S}_{i+1} &= \boldsymbol{S}_i + \sum_{m=1}^{M_{i+1}} Pr\left(k|\boldsymbol{o}_m, \lambda_{UBM}\right) \boldsymbol{o}_m^2
\end{aligned}
\tag{5.1}
$$

The mean, variance and weights of the updated model $s^{(i+1)}$ are then obtained according to Equation (4.2) in Chapter 4, Section 4.1.1. This procedure is linear and

thus, given the same data, the incremental MAP procedure will produce the same models as the off-line procedure, while still being suited to on-line processing.

5.2.2 System implementation

The proposed semi-supervised on-line diarization system is illustrated in Fig. 5.5. It is based on the baseline top-down or divisive hierarchical clustering approach to off-line diarization reported in Chapter 3, Section 3.4 and the unsupervised on-line diarization approach described in Chapter 4.

The system is characterised by four stages: (i) feature extraction; (ii) off-line speaker models enrolment; (iii) speech activity detection and (iv) on-line classification.

Feature extraction

Each audio stream is first parametrised by a series of acoustic observations $\mathbf{o}_1, \ldots, \mathbf{o}_T$. Critically, for any time $\tau \in 1, \ldots, T$ only those observations for $t < \tau$ are used for diarization.

Off-line speaker models enrolment

A brief round-table phase in which each speaker introduces himself is used to seed speaker models. The first T_{SPK} seconds of active speech for each speaker is set aside as seed labelled training data. An inventory $\tilde{\Delta}$ of speaker models s_j, where $j = 1, \ldots, M$, with M indicating the number of speakers in any particular meeting, is then trained using a certain duration of seed data T_{SPK} for each speaker. Speaker models are MAP adapted from the UBM using the seed data. For each speaker model s_j, the sufficient statistics $N_1^{(j)}, \boldsymbol{F}_1^{(j)}$ and $\boldsymbol{S}_1^{(j)}$ obtained during the MAP adaptation are stored in order to be used during the on-line classification phase to update the speaker models. The resulting set of seed speaker models are then used to diarize the remaining speech segments in an unsupervised fashion.

Speech activity detection and on-line classication

Non-speech segments are removed according to the output of a conventional model-based speech activity detector (SAD) derived from the baseline top-down diarization system described in Chapter 3, Section 3.4.

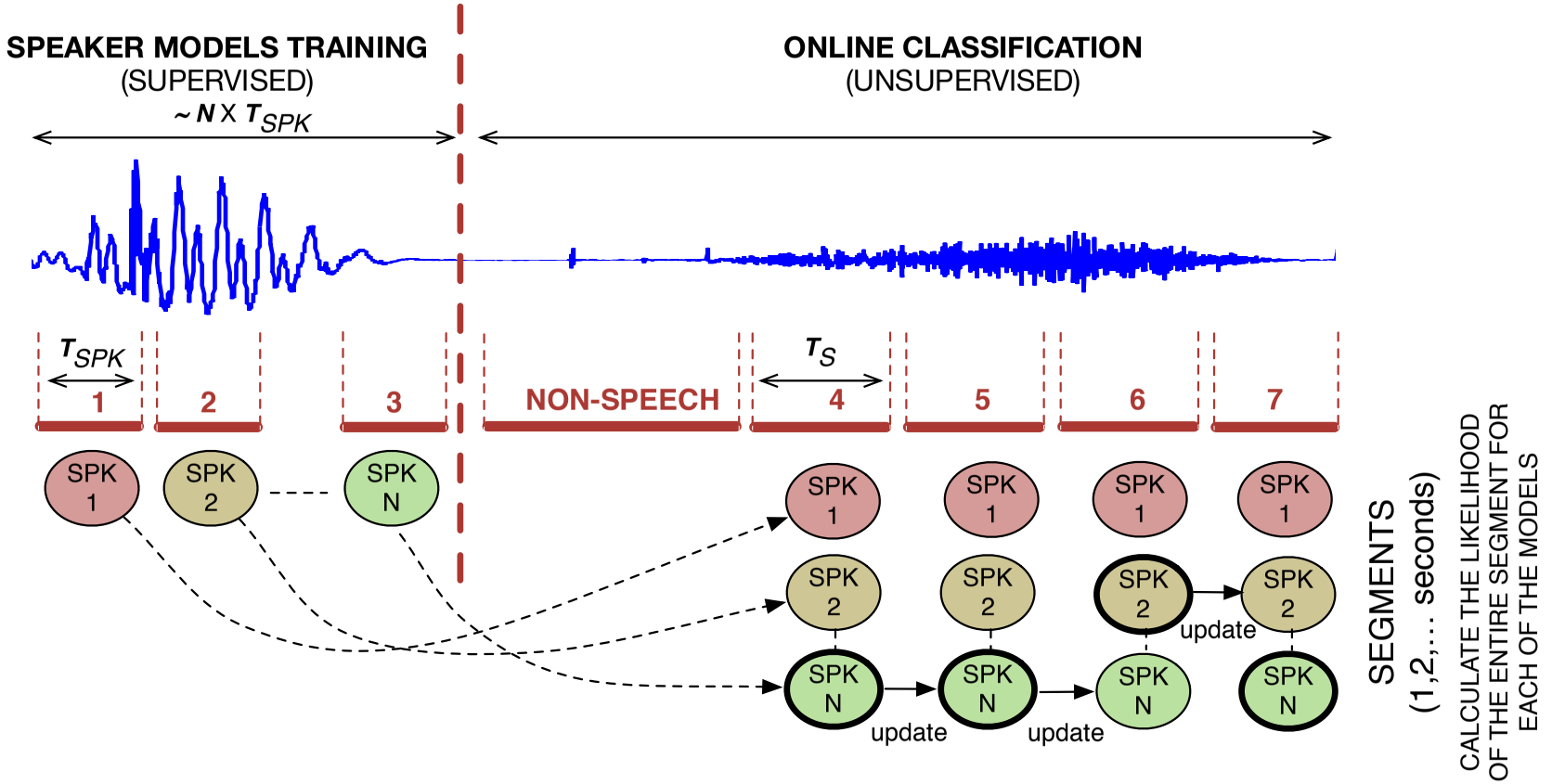


Fig. 5.5 An illustration of the semi-supervised on-line speaker diarization system.

The remaining speech segments are then divided into smaller sub-segments whose duration is no longer than an a-priori fixed maximum duration T_S. Higher values of T_S imply a higher latency system. On-line diarization is then applied in sequence to each sub-segment. The optimised speaker sequence $\tilde{S}$ and segmentation $\tilde{G}$ are obtained by assigning in sequence each segment i to one of the M speaker models according to:

$$s_j = \arg\max_{l \in (1,\ldots,M)} \sum_{k=1}^{K} \mathcal{L}\left(\mathbf{o}_k | s_l\right) \tag{5.2}$$

where $\mathbf{o}_k$ is the k-th acoustic feature in the segment i, K represents the number of acoustic features in the i-th segment and where $\mathcal{L}\left(\mathbf{o}_k | s_l\right)$ denotes the log-likelihood of the k-th feature in segment i given the speaker model s_l. The segment is then labelled according to the recognised speaker j as per (5.2). The updated speaker model s_j is obtained by either sequential or incremental MAP adaptation as described above.

5.3 Performance evaluation

In order to evaluate the performance of the proposed on-line semi-supervised diarization system, average global DERs are assessed as a function of different amount of labelled training data T_{SPK} and for different maximum segment duration T_S. The evaluation aims to determine what quantity of manually labelled seed data is needed to obtain the performance of the state-of-the-art, entirely off-line, baseline system reported in Chapter 3, Section 3.4, the associated cost in terms of system latency and the benefit of incremental MAP adaptation.

5.3.1 Semi-supervised on-line diarization against off-line diarization performance

First, during the off-line and supervised phase, speaker models s_j are trained using increasing quantities of labelled training data T_{SPK} of duration $1, \ldots, 39$ seconds. The general UBM model of 64 Gaussian components is the same as the one used for ASV experiments.

Since there is no round-table phase in the RT data, this component is simulated. Seed data is taken from wherever the first T_{SPK} seconds of speaker data are found. This

T_{SPK}	3 sec.		5 sec.		7 sec.	
Algo. MAP	**Seq**	**Inc**	**Seq**	**Inc**	**Seq**	**Inc**
RTdev	24.7	21.3	21	18.1	20.5	16.5
RT07	19.1	17.3	17.5	14.6	13.6	13.3
RT09	23.7	18.2	17.6	16.2	21.2	16.2
Average	22.4	**18.9**	18.7	**16.3**	18.5	**15.3**

Seq = Sequential MAP; **Inc** = Incremental MAP

Table 5.1 A comparison of DER using sequential and incremental MAP algorithms. Results are reported for a segment duration / latency T_S of 3 seconds, three different datasets RTdev, RT07 and RT09 and for different durations T_{SPK} of training data.

has only a negligible bearing on the subsequent assessment of diarization performance, as the majority of speakers speak for more than 2 minutes.

On-line diarization is performed using different maximum segment duration T_S with $T_S = 0.25, 0.5, 1, 2, 3, 4$. Greater values of T_S imply a higher latency of the system. Performance of the system is evaluated both using the standard sequential and incremental MAP adaptation procedure in order to prove the better efficiency of the latter in delivering lower diarization error rate.

Results in Figures 5.7, 5.8 and 5.9 illustrate the variation in DER against the amount of speaker training data T_{SPK} for the RTdev, RT07 and RT09 datasets respectively. Left plots illustrate performance for sequential MAP adaptation whereas right plots correspond to incremental MAP adaptation. In each plot, different profiles illustrate performance for a range of segment durations / latencies T_S.

The first observation from Figures 5.7, 5.8 and 5.9 indicates that the performance of the semi-supervised, on-line diarization system can surpass that of the baseline, off-line diarization system (illustrated with horizontal, dashed lines). In the case of sequential MAP adaptation this is achieved for the RTdev dataset, for instance, when speaker models are seeded with $T_{SPK} = 9$ seconds of training data when using a segment size / latency of $T_S = 4$ seconds. With the same segment size, the baseline performance for the RT07 and RT09 datasets is surpassed using as little as $T_{SPK} = 5$ and 3 seconds respectively.

In general, lower DERs are achieved with greater quantities of seed data, for instance a DER of 12.5% is achieved with $T_{SPK} = 9$ seconds of training data for the RT07

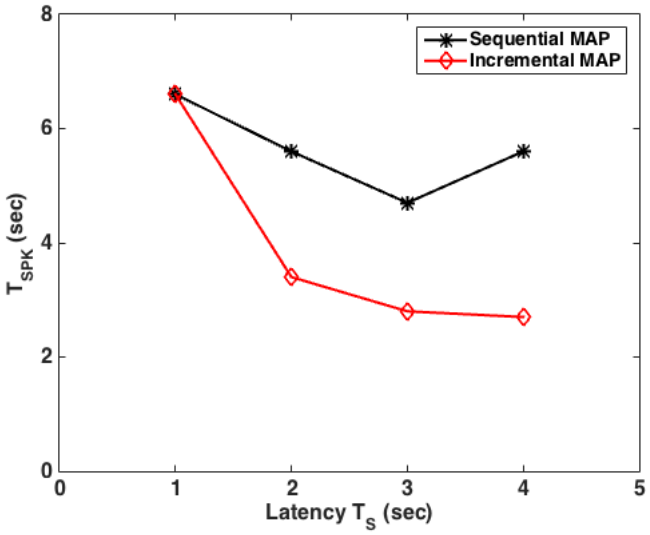


Fig. 5.6 Speaker training data duration T_{SPK} against segment duration / latency T_S for the RT07 evaluation dataset using sequential and incremental MAP algorithms. All points correspond to a DER of 18 % (baseline, off-line performance).

dataset and 15% with 17 seconds of training data for the RT09 dataset, both with latencies of $T_S = 3$ seconds.

Turning next to results for incremental MAP illustrated in the right plots of Figures 5.7, 5.8 and 5.9 it is immediately evident that performance is significantly better than for sequential MAP. Here, the baseline, off-line diarization performance is surpassed with as little as $T_{SPK} = 5$ seconds of seed data for the RTdev dataset and $T_{SPK} = 3$ seconds in the case of both RT07 and RT09, all with a latency as low as $T_S = 2$ seconds. Once again, lower DERs are achieved with greater quantities of seed data, as low as 10% for the RT07 dataset and 12.5% for the RT09 dataset.

Table 5.1 summaries results across the three different datasets for T_{SPK}=3, 5, and 7 seconds of speaker training data and a fixed latency of $T_S = 3$ seconds. Results are illustrated for sequential and incremental MAP adaptation algorithms whereas average performance is illustrated in the bottom row. In all cases, incremental MAP adaptation delivers lower DERs.

Figure 5.6 plots the quantity of speaker training data T_{SPK} as a function of the latency T_S for the evaluation dataset RT07. All points correspond to a DER of 18% and thus show different configurations which achieve the same performance as the baseline, off-line diarization system. Plots are illustrated for both sequential and incremental

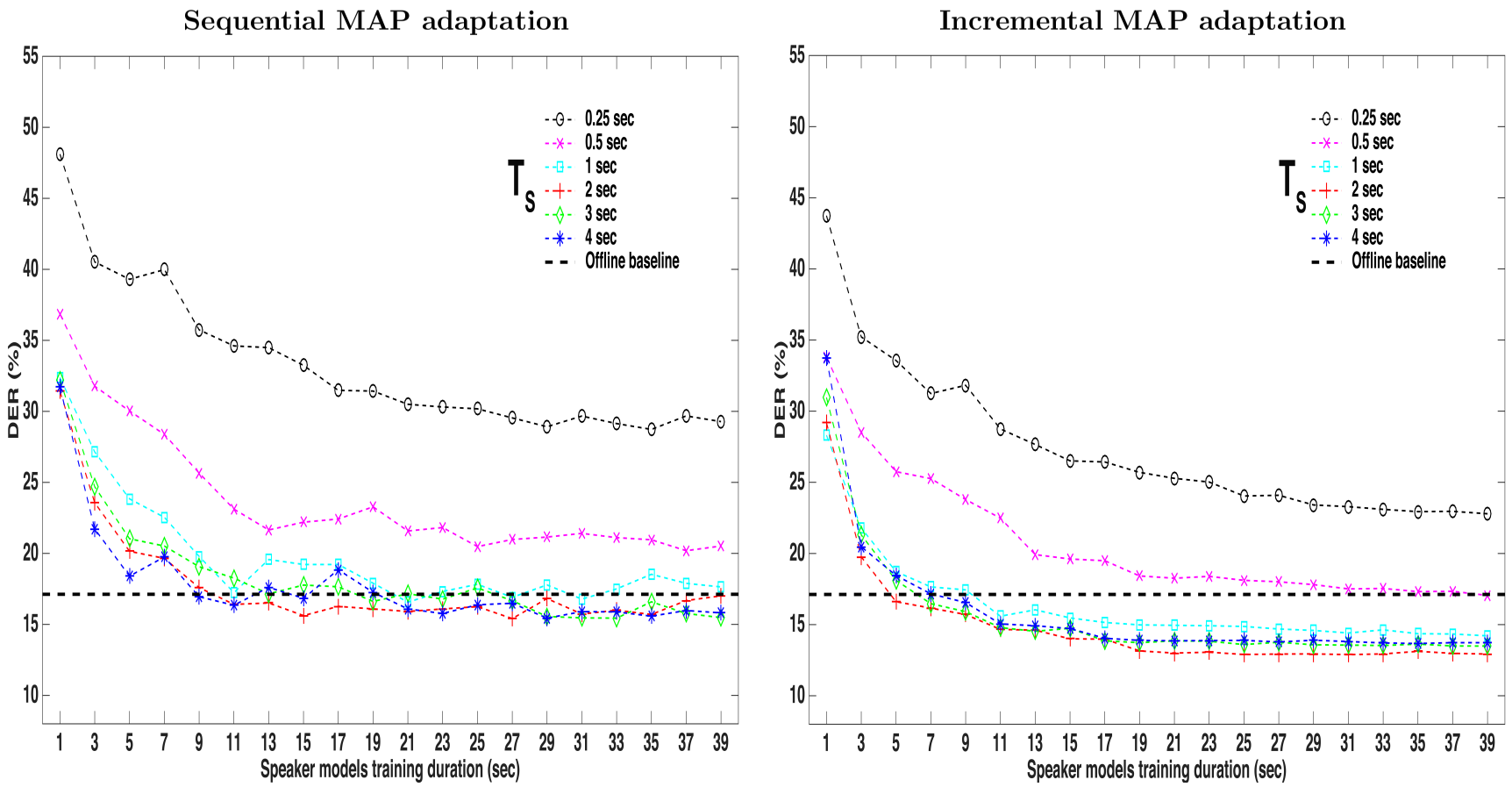


Fig. 5.7 An illustration of DER for the semi-supervised on-line diarization system as a function of the speaker model training duration T_{SPK} and for different maximum segment durations / latency T_S. Results shown for the RTdev development dataset using sequential MAP adaptation (left) and incremental MAP adaptation (right). The horizontal, dashed line indicates the performance of the baseline, off-line diarization system.

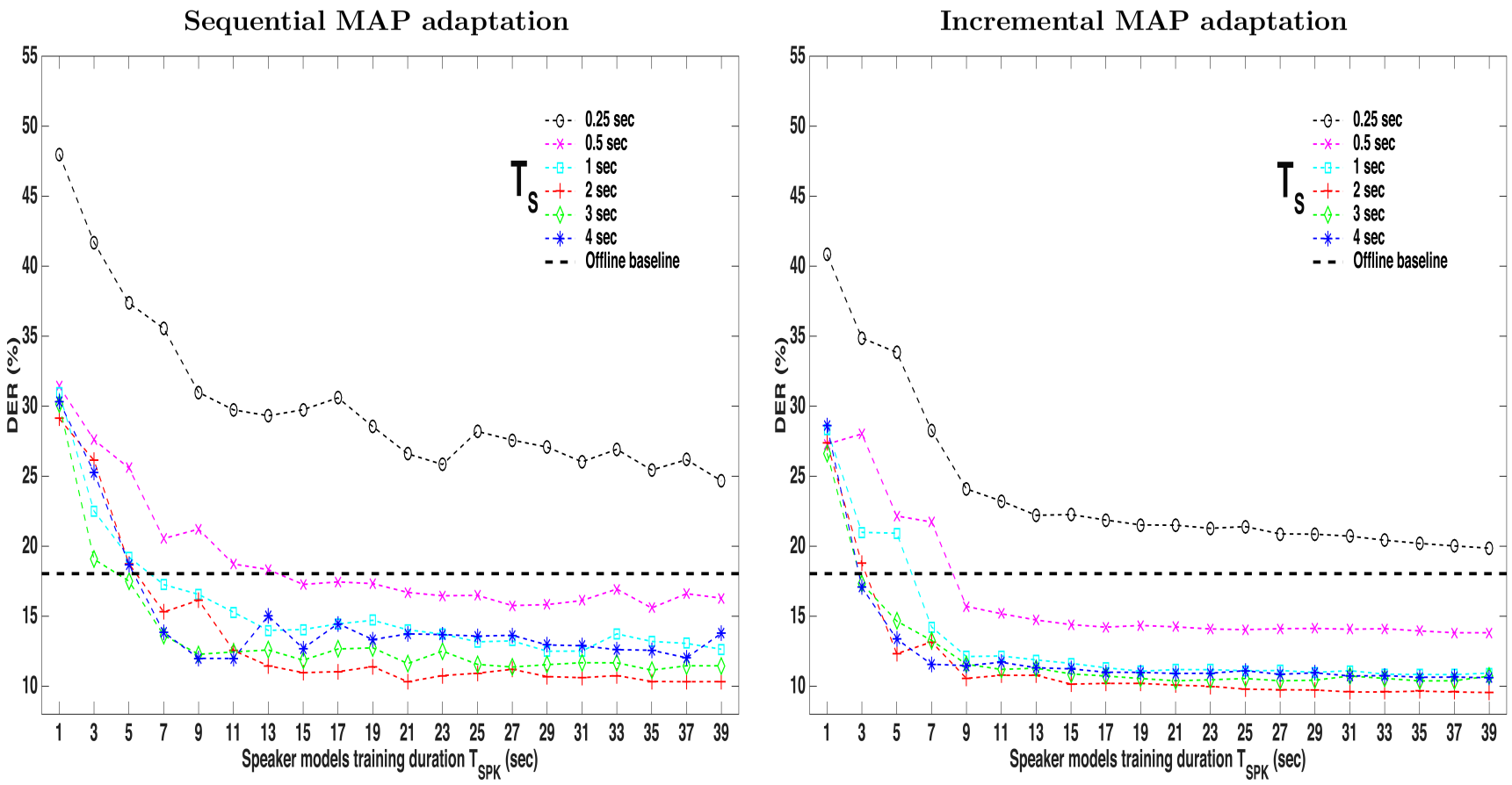


Fig. 5.8 An illustration of DER for the semi-supervised on-line diarization system as a function of the speaker model training duration T_{SPK} and for different maximum segment durations / latency T_S. Results shown for the RT07 evaluation dataset using sequential MAP adaptation (left) and incremental MAP adaptation (right). The horizontal, dashed line indicates the performance of the baseline, off-line diarization system.

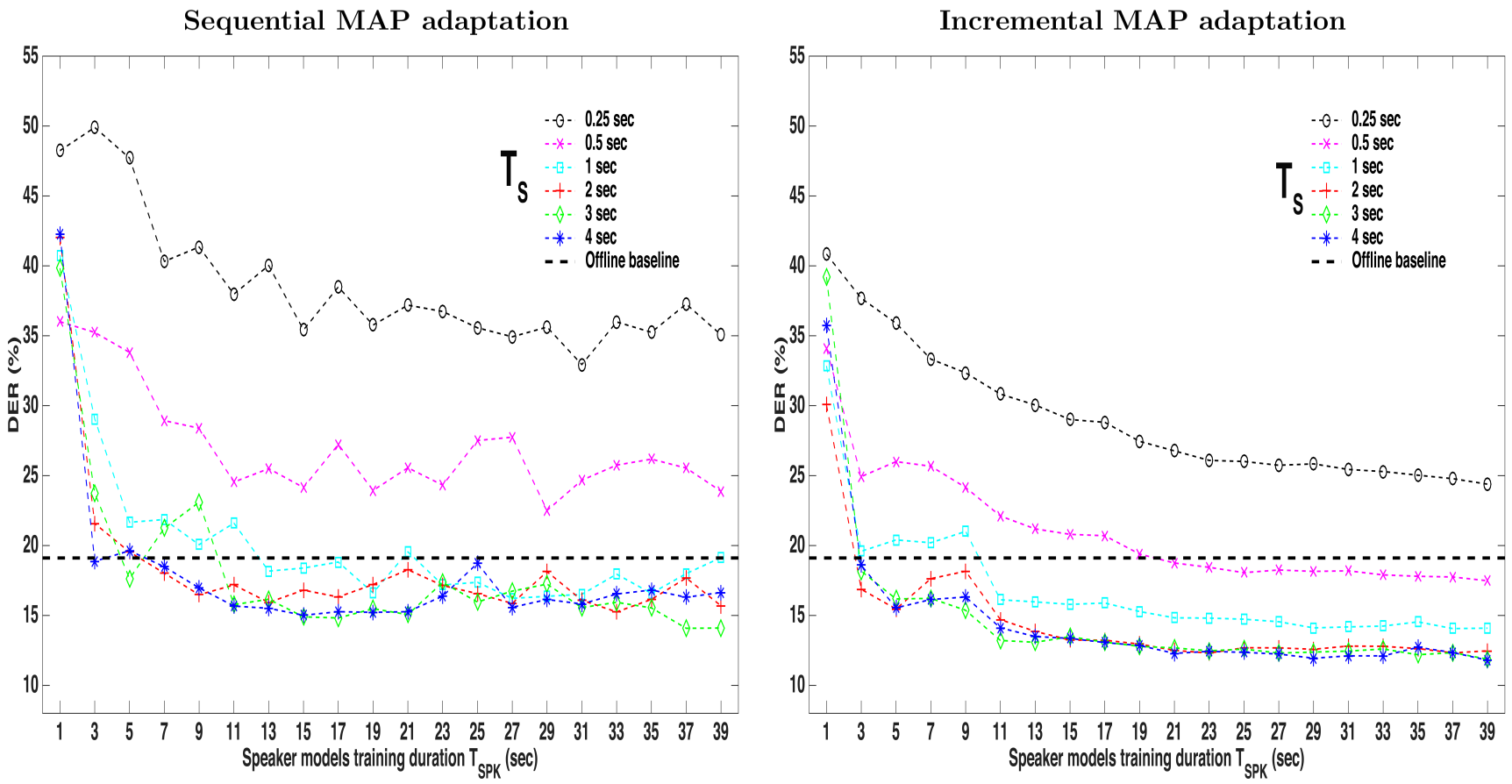


Fig. 5.9 An illustration of DER for the semi-supervised on-line diarization system as a function of the speaker model training duration T_{SPK} and for different maximum segment durations / latency T_S. Results shown for the RT09 evaluation dataset using sequential MAP adaptation (left) and incremental MAP adaptation (right). The horizontal, dashed line indicates the performance of the baseline, off-line diarization system.

MAP adaptation algorithms. In all cases, incremental MAP adaptation matches or betters baseline, off-line diarization performance with a lower amount of seed data or a lower latency than sequential MAP adaptation.

Finally, results presented in Figures 5.7, 5.8 and 5.9 indicate that values of $T_S > 1$ seconds of latency are required for the best performance, no matter what is the value of T_{SPK}. Performance degrades universally for lower latencies. Crucially, for all datasets, DERs are equivalent or better than that of the baseline, off-line system when $T_S > 0.5$ seconds and when given sufficient training data T_{SPK}.

5.4 Summary

This chapter reports our efforts to improve the unsupervised on-line diarization system presented in Chapter 4 and presents a semi-supervised on-line diarization system able to reach the performance of a state-of-the-art off-line diarization system. The relaxation of supervision constraints overcomes the difficulty in initialising speaker models in an unsupervised fashion with small quantities of data; the use of longer segments would come at the expense of increased system latency.

In the case of the RT07 evaluation dataset, it is shown that such a system can outperform an off-line diarization system with just 3 seconds of speaker seed data and 3 seconds of latency when using an incremental MAP adaptation procedure. By using greater quantities of seed data or by allowing greater latency, then a diarization error rate in the order of 10% can be achieved.

While these levels of performance may support practical applications, the need for supervised training remains an inconvenience. If the inconvenience of a short, initial training phase proves acceptable, then this opens the potential for the application of either supervised or semi-supervised speaker discriminant feature transformations which may offer an opportunity for improved performance. This work could reduce the need for seed data, latency, or both. One avenue through which this objective might be pursued involves the application of phone adaptive training (PAT), a technique to marginalise the phonetic variation in short duration sentences which is introduced and described in the next Chapter 6.

Chapter 6

Phone adaptive training

Many automatic speech processing applications, such as the unsupervised and semi-supervised on-line speaker diarization systems proposed in Chapter 4 and 5 and short-duration text-independent automatic speaker verification (ASV) systems [65–67], are required to operate in the face of varying data quantities during the speaker modelling phase. When data is plentiful, nuisance or phonetic variation can be implicitly normalised or marginalised and often has limited or no impact on performance. For example, the use of long-duration training and testing data in ASV systems effectively compensates for the effect of differing phone content. In contrast, when training data is scarce, then speaker models are biased towards the specific phonetic content and performance can degrade drastically if the phonetic variation is dissimilar to that encountered in testing; phonetic variation is no longer marginalised.

Phone adaptive training (PAT) is a recently introduced algorithm [4] whose aim is to normalise phone variation in the scope of speaker diarization by projecting acoustic features into a new space in which phone discrimination is minimised while speaker discrimination is maximised. PAT is based on the original idea of speaker adaptive training (SAT) [68], a technique commonly used in automatic speech recognition (ASR) and language recognition [69, 70]. SAT projects speaker-dependent features into a speaker-neutral space in order that recognition may be performed reliably using speaker-independent models. By interchanging, the role of phones and speakers, PAT suppresses phone variation while emphasising speaker variation. While PAT operates at the feature level and targets improved speaker modelling, its use within a speaker diarization framework makes for somewhat troublesome optimisation.

The first contribution of this chapter is the assessment and optimisation of PAT in isolation from the convolutive complexities of speaker diarization and under strictly controlled conditions. By means of oracle ASV experiments, PAT performance is analysed when applied to short-duration text-independent ASV as a function of model complexity and for varying quantities of training data, using the TIMIT dataset, described in Chapter 3, Section 3.5, which is manually labelled at the phone level.

The second contribution of this chapter consists of our efforts to develop PAT into a fully unsupervised system. Contributions include an approach to automatic acoustic class transcription using regression tree analysis. Similarly to the first work, the performance of PAT is analysed as a function of model complexity and for varying quantities of training data. Experiments show that PAT performs well even when the number of acoustic classes is reduced well below the number of phones thus reducing the need of accurate ground-truth phonetic transcriptions.

This chapter is organised as follows. Section 6.1 outlines previous related work. Section 6.2 describes the SAT technique and introduces PAT algorithm. Section 6.3 describes oracle ASV experiments performed to assess PAT performance in optimal conditions. Section 6.4 describes the ASV experiments when using automatic acoustic class transcriptions. A brief summary is provided in Section 6.5.

6.1 Prior work

The influence of phone variation in degrading the performance of short duration speaker recognition and speaker diarization is well acknowledged [71, 67, 72, 73]. The work in [73] illustrates that, as the quantity of data used for model training is reduced, then the phone distribution tends to be more and more dissimilar. Fauve et al. [65] analysed the impact of short-duration training utterances on two automatic speaker verification (ASV) systems: a Gaussian mixture model system with a universal background model (GMM-UBM) and a GMM supervector system based on a support vector machine (SVM) classifier. They showed that conventional inter-session compensation techniques and ASV attain sub-optimal performance when confronted with short-duration training utterances. The same authors highlighted in [74] the sensitivity of speech activity detection (SAD) and the limitations of maximum a posteriori (MAP) adaptation in the case of short-duration training. Eigenvoice modelling was shown to improve robustness

by removing model components which are insufficiently adapted as a result of training data scarcity.

Other authors have investigated the impact of duration mis-match, namely differences in the data quantities used for modelling and testing. In the context of a joint factor analysis (JFA) system, Vogt et al. [66] showed that ASV performance degrades when speaker and channel sub-spaces are trained on full-length utterances, but short utterances are used for testing. This behaviour is caused by the phone-variation in short utterances which tends to dominate the effects of inter-session variability. Improved ASV performance was obtained by training the channel subspace matrix on utterances of duration similar to those encountered during testing. Other work reported in [67, 72, 73, 75] showed similar effects on iVector [76] and probabilistic linear discriminant analysis (PLDA) [41] system variants. Common to all this work is the modelling and testing using similar quantities of data, thereby marginalising to some extent the effects of phone-variation.

The work in [77–79] all investigated approaches to compensate for phone variation in the context of speaker identification (SI). That in [77] investigated the projection of features into a phone-independent subspace in order to improve text-independent SI. Based on the assumption that phone variation dominates speaker variation, the phone-independent subspace is learned using principal component analysis (PCA). Features are then projected onto the eigenvectors which correspond to the lowest eigenvalues. In [78] probabilistic principal component analysis (PPCA) is used instead to learn the phoneme-independent subspace.

Other more generalised techniques such as maximum likelihood linear regression (MLLR) and constrained maximum likelihood linear regression (cMLLR), have been used extensively to improve speaker discriminability and thus to improve ASV performance. Stolcke et al. [80] and Ferras et al. [81] both used speaker dependent MLLR and cMLLR transforms in order to model the difference between speaker independent and dependent models. The estimated transforms capture speaker dependent characteristics and are used themselves as features in order to train SVM-based verification systems. With the aid of ASR transcripts, these approaches can exploit knowledge of the phone content in order to estimate phone-neutral speaker models [80]. Stolcke's later work [82] showed that the same phone content can be used to derive more speaker-discriminative

cMLLR transforms using speech-constrained phonetic regions defined by prosodic and phonetic criteria.

When training data is scarce, for instance in the case of short-duration ASV or in the case of model initialization in some approaches to speaker diarization, the learning of speaker and phone specific transforms can be impractical. Phone adaptive training (PAT), introduced by Bozonnet et al. [4], differs from the previous work in that phone-dependent cMLLR transforms are learned in a speaker-independent fashion. PAT is used to project acoustic features into a new, phone-normalised space which is more discriminative in terms of speakers. Of particular appeal, the projected features can be used in the place of baseline features with any ASV or diarization system.

6.2 From SAT to PAT

This section provides a description of maximum likelihood linear regression (MLLR) and constrained maximum likelihood linear regression (cMLLR), two model adaptation techniques which form the basis of SAT and PAT. A thorough explanation of the SAT technique for speech recognition from which is derived PAT is provided. Finally, the PAT algorithm for phonetic normalisation is presented.

6.2.1 MLLR

Maximum likelihood linear regression (MLLR) is an affine transform approach to model adaptation. The aim is to reduce the mismatch between a model and an adaptation dataset. As detailed in [83, 84], when the model is a GMM with initial model mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, then the adapted mean $\hat{\boldsymbol{\mu}}$ and covariance $\hat{\boldsymbol{\Sigma}}$ are estimated according to:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b} \tag{6.1}$$

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{B}\boldsymbol{H}\boldsymbol{B}^T \tag{6.2}$$

where the transform is characterized by an $n \times n$ regression matrix $\boldsymbol{A}$ (n being the dimension of the feature space), an n-dimensional bias vector $\boldsymbol{b}$ and an $n \times n$ matrix

$\boldsymbol{H}$. $\boldsymbol{B}$ is the inverse of the Cholesky factor $\boldsymbol{C}$ of $\boldsymbol{\Sigma}^{-1}$:

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{C}\boldsymbol{C}^T \tag{6.3}$$

$$\boldsymbol{B} = \boldsymbol{C}^{-1} \tag{6.4}$$

Both $\boldsymbol{A}$ and $\boldsymbol{b}$ are optimized according to a standard expectation maximisation (EM) algorithm [26] to maximize the likelihood of the model with respect to the adaptation data.

6.2.2 cMLLR

In contrast to standard MLLR, which requires two different, independently optimised transforms, $(\boldsymbol{A}, \boldsymbol{b})$ and $\boldsymbol{H}$, the constrained MLLR (cMLLR) algorithm requires a single transform $\boldsymbol{W} = (\boldsymbol{A}, \boldsymbol{b})$ to adapt both mean and variance parameters [85]. Equations 6.1 and 6.2 then become:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b} \tag{6.5}$$

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T \tag{6.6}$$

where the transform $\boldsymbol{A}$ and $\boldsymbol{b}$ are the constrained $n \times n$ transform matrix and the n-dimensional bias vector respectively, both still estimated in the maximum likelihood sense from the training data. Since the mean and the variance transforms are tied, in addition to model transformation, cMLLR can also be used to transform an acoustic feature $\boldsymbol{o}$ according to:

$$\hat{\boldsymbol{o}} = \boldsymbol{A}^{-1}\boldsymbol{o} - \boldsymbol{A}^{-1}\boldsymbol{b} \tag{6.7}$$

The application of cMLLR at the feature level is the starting point for SAT and PAT.

6.2.3 SAT

Speaker adaptive training (SAT) is a technique mainly used in automatic speech recognition (ASR) applications in order to remove the speaker component while

retaining the relevant phonetic information. As illustrated into Figure 6.1 the original acoustic features coming from different speakers are projected in a new acoustic space where the phonetic discrimination is higher while speaker discrimination is minimised.

Mathematically, we suppose a speech dataset of utterances collected from S different speakers ($s = 1, 2, \ldots, S$) parametrized by a set of acoustic features $\boldsymbol{O}$ represented by $\boldsymbol{O}_s = (\boldsymbol{o}_{s,1}, \ldots, \boldsymbol{o}_{s,N_s})$ where N_s is the number of acoustic features corresponding to each speaker $s \in S$. Assuming that all features are characterised by the same speaker, source, channel and noise level conditions, then an optimal acoustic model $\boldsymbol{\lambda}_{opt}$ can be estimated according to the following equation:

$$\boldsymbol{\lambda}_{opt} = \arg\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{O}|\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{\lambda}} \prod_{s=1}^{S} \mathcal{L}(\boldsymbol{O}_s|\boldsymbol{\lambda}) \tag{6.8}$$

with $\mathcal{L}(\boldsymbol{O}_s|\boldsymbol{\lambda})$ being the likelihood of the model $\boldsymbol{\lambda}$ with respect to the acoustic observations $\boldsymbol{O}_s$.

However speech recognition is usually negatively affected by the variation due to different speakers. For each speaker s, SAT estimates iteratively a transformation $\tilde{\boldsymbol{W}}_s = (\tilde{\boldsymbol{A}}_s, \tilde{\boldsymbol{b}}_s)$ which captures the speaker variability. Simultaneously, SAT learns a speaker-independent acoustic model $\boldsymbol{\lambda}_c$ which captures the phonetic information with which ASR can be performed reliably.

The algorithm to calculate $\boldsymbol{\lambda}_c$ is thus defined by:

$$\left(\boldsymbol{\lambda}_c, \tilde{\boldsymbol{W}}\right) = \arg\max_{\boldsymbol{\lambda}, \boldsymbol{W}} \prod_{s=1}^{S} \mathcal{L}\left(\boldsymbol{O}_s|\boldsymbol{W}_s, \boldsymbol{\lambda}\right) \tag{6.9}$$

where $\tilde{\boldsymbol{W}} = (\tilde{\boldsymbol{W}}_1, \ldots, \tilde{\boldsymbol{W}}_s)$ represents the set of speaker transforms. As in Equation 6.7, speaker-normalized features $\tilde{\boldsymbol{O}}_s$ are then obtained according to:

$$\tilde{\boldsymbol{o}}_{s,t} = \tilde{\boldsymbol{A}}_s^{-1} \boldsymbol{o}_{s,t} - \tilde{\boldsymbol{A}}_s^{-1} \tilde{\boldsymbol{b}}_s \tag{6.10}$$

where $t = 1, \ldots, N_s$ is the feature index. Since there is no closed-form solution, Equation 6.9 is optimised iteratively.

We denote by $\boldsymbol{O}_s^{(0)}$ the set of initial acoustic feature vectors for each speaker s. The initial step consists in training a speaker model $\boldsymbol{\lambda}_c^{(0)}$ using the initial acoustic feature vectors. Then, for each iteration i, the algorithm, illustrated in Figure 6.2, proceeds as follows:

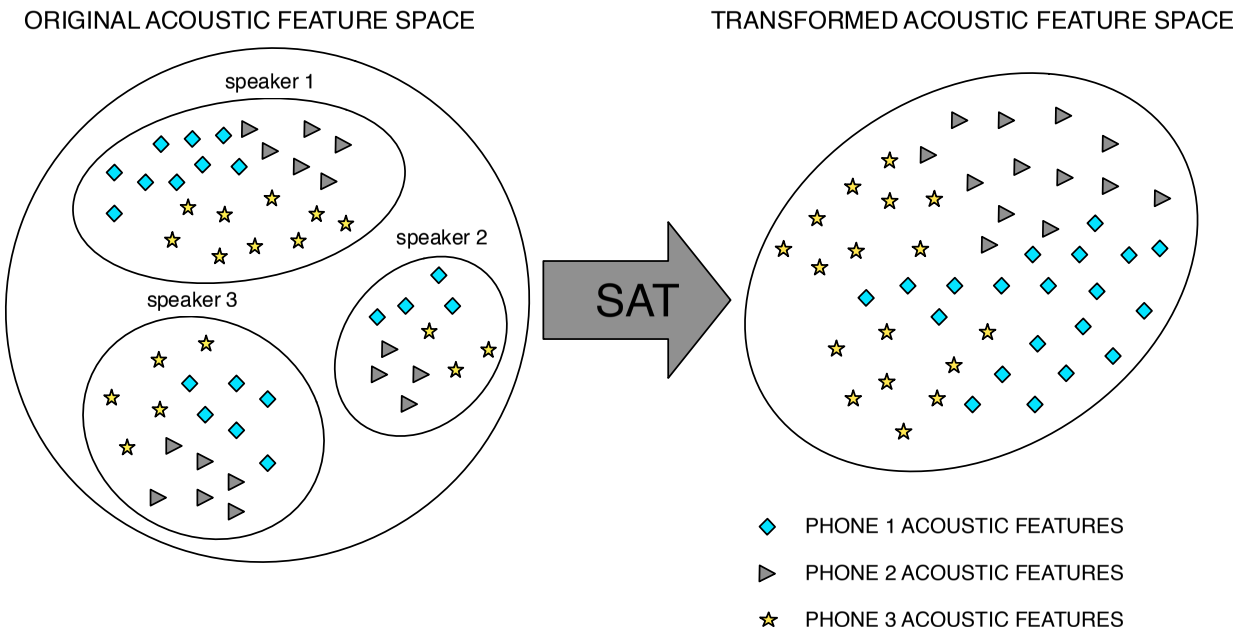


Fig. 6.1 An illustration of SAT.

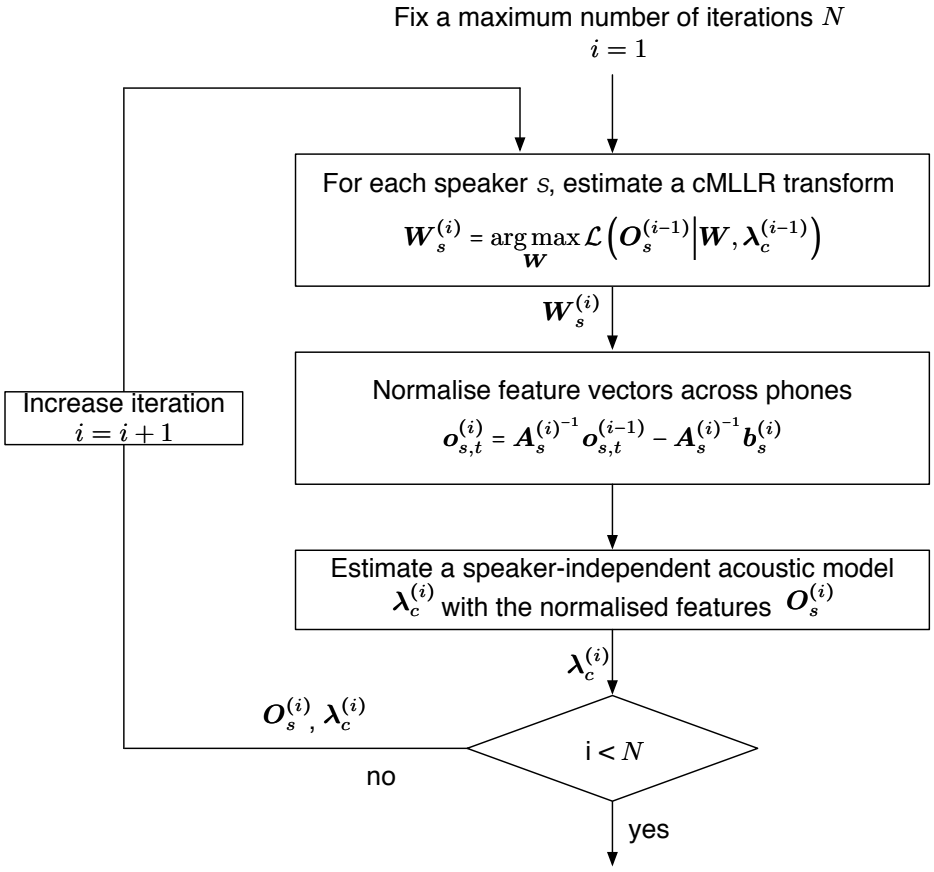


Fig. 6.2 An illustration of the SAT algorithm.

1. Estimate a cMLLR transform $\boldsymbol{W}_s^{(i)} = (\boldsymbol{A}_s^{(i)}, \boldsymbol{b}_s^{(i)})$ for each speaker s such that:

$$\boldsymbol{W}_s^{(i)} = \arg\max_{\boldsymbol{W}} \mathcal{L}\left(\boldsymbol{O}_s^{(i-1)} \middle| \boldsymbol{W}, \boldsymbol{\lambda}_c^{(i-1)}\right) \tag{6.11}$$

2. Apply the transform $\boldsymbol{W}_s^{(i)}$ obtained in step 1 to the set of acoustic features resulting from iteration $i-1$ to obtain a new set of speaker-normalised acoustic features for each speaker s:

$$\boldsymbol{o}_{s,t}^{(i)} = \boldsymbol{A}_s^{(i)^{-1}} \boldsymbol{o}_{s,t}^{(i-1)} - \boldsymbol{A}_s^{(i)^{-1}} \boldsymbol{b}_s^{(i)} \tag{6.12}$$

3. Retrain the speaker-independent acoustic model $\boldsymbol{\lambda}^{(i-1)}$ obtained at step $i-1$, estimate a new set of normalised speaker models $\boldsymbol{\lambda}_c^{(i)}$ for each speaker s, using the speaker-normalised acoustic features $\boldsymbol{O}_s^{(i)}$ obtained in step 2.

4. Increase i to $i+1$ and iterate from step 1 until a maximum number of iterations is reached.

For each speaker s, the final iteration produces speaker-normalised acoustic features $\tilde{\boldsymbol{O}}_s$, cMLLR speaker transforms $\tilde{\boldsymbol{W}}_s$ and a speaker-independent acoustic model $\boldsymbol{\lambda}_c$.

6.2.4 PAT

The motivation of PAT stems from the idea behind SAT. As illustrated in Figure 6.3, PAT aims to project acoustic features into a space where phone variability is suppressed in order to provide more speaker-discriminative features for speaker modelling.

We suppose a dataset of utterances collected from S different speakers. Each utterance is composed of P different phones such that the global set of acoustic features is represented by $\boldsymbol{O}_{s,p} = (\boldsymbol{o}_{s,p,1}, \ldots, \boldsymbol{o}_{s,p,N_{s,p}})$ where $N_{s,p}$ is the number of acoustic features corresponding to each speaker $s \in S$ and each phone $p \in P$. For each phone p, PAT estimates iteratively a transformation $\tilde{\boldsymbol{W}}_p = (\tilde{\boldsymbol{A}}_p, \tilde{\boldsymbol{b}}_p)$ which captures the phone variation across speakers. Simultaneously, PAT learns a set of phone-normalised speaker models $\tilde{\boldsymbol{\Lambda}} = (\tilde{\boldsymbol{\lambda}}_1, \ldots, \tilde{\boldsymbol{\lambda}}_S)$. The algorithm is thus defined by:

$$\left(\tilde{\boldsymbol{\Lambda}}, \tilde{\boldsymbol{W}}\right) = \arg\max_{\boldsymbol{\Lambda}, \boldsymbol{W}} \prod_{s=1}^{S} \prod_{p=1}^{P} \mathcal{L}\left(\boldsymbol{O}_{s,p} | \boldsymbol{W}_p, \boldsymbol{\lambda}_s\right) \tag{6.13}$$

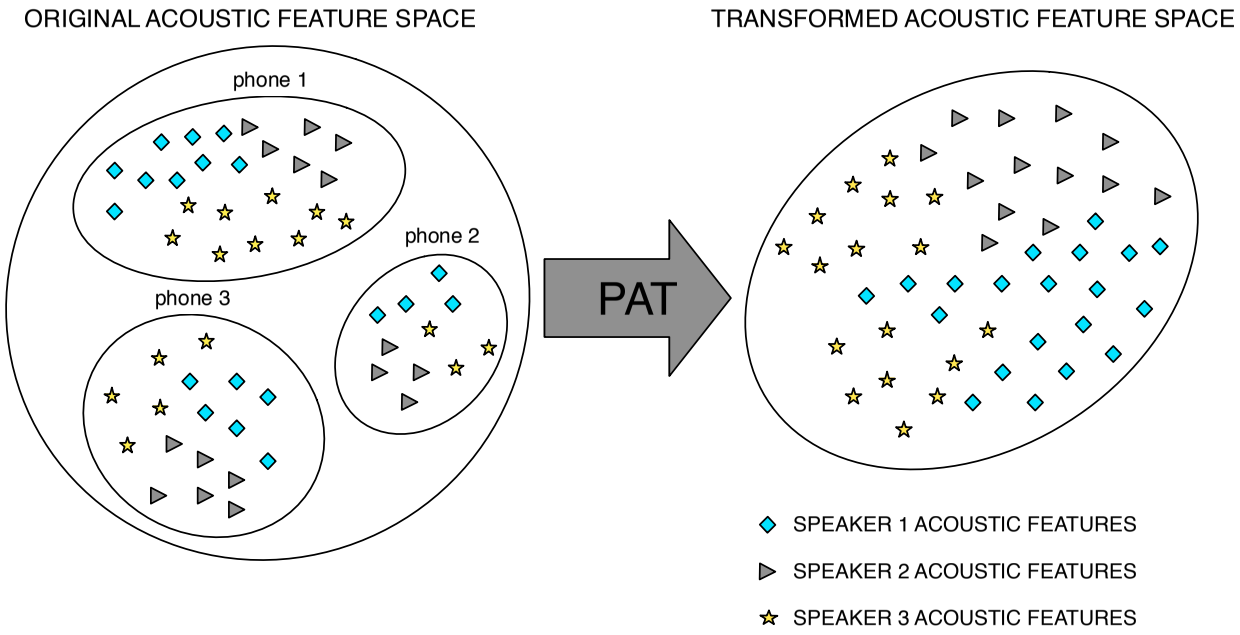


Fig. 6.3 An illustration of PAT.

where $\tilde{\boldsymbol{W}} = (\tilde{\boldsymbol{W}}_1, \ldots, \tilde{\boldsymbol{W}}_p)$ represents the set of phone transforms. As in Equation 6.7, phone-normalized features $\tilde{\boldsymbol{O}}_{s,p}$ are then obtained according to:

$$\tilde{\boldsymbol{o}}_{s,p,t} = \tilde{\boldsymbol{A}}_p^{-1} \boldsymbol{o}_{s,p,t} - \tilde{\boldsymbol{A}}_p^{-1} \tilde{\boldsymbol{b}}_p \tag{6.14}$$

where $t = 1, \ldots, N_{s,p}$ is the feature index. Since there is no closed-form solution, Equation 6.13 is optimised iteratively.

We denote by $\boldsymbol{O}_{s,p}^{(0)}$ the set of initial acoustic feature vectors for each speaker s and phone p. The initial step consists in training a set of speaker models $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1^{(0)}, \ldots, \boldsymbol{\lambda}_S^{(0)})$ using the initial acoustic features vectors. Then, for each iteration i, the algorithm, illustrated in Figure 6.4, proceeds as follows:

1. Estimate a cMLLR transform $\boldsymbol{W}_p^{(i)} = (\boldsymbol{A}_p^{(i)}, \boldsymbol{b}_p^{(i)})$ for each phone p such that:

$$\boldsymbol{W}_p^{(i)} = \arg\max_{\boldsymbol{W}} \prod_{s=1}^{S} \mathcal{L}\left(\boldsymbol{O}_{s,p}^{(i-1)} \middle| \boldsymbol{W}, \boldsymbol{\lambda}_s^{(i-1)}\right) \tag{6.15}$$

2. Apply the transform $\boldsymbol{W}_p^{(i)}$ obtained in step 1 to the set of acoustic features resulting from iteration $i-1$ to obtain a new set of phone-normalised acoustic features for each speaker s and phone p:

$$\boldsymbol{o}_{s,p,t}^{(i)} = \boldsymbol{A}_p^{(i)^{-1}} \boldsymbol{o}_{s,p,t}^{(i-1)} - \boldsymbol{A}_p^{(i)^{-1}} \boldsymbol{b}_p^{(i)} \tag{6.16}$$

3. Through MAP adaptation of speaker models $\boldsymbol{\Lambda}^{(i-1)}$ obtained at step $i-1$, estimate a new set of normalised speaker models $\boldsymbol{\Lambda}^{(i)} = (\boldsymbol{\lambda}_1^{(i)}, \ldots, \boldsymbol{\lambda}_S^{(i)})$ for each speaker s, using the phone-normalised acoustic features $\boldsymbol{O}_{s,p}^{(i)}$ obtained in step 2.

4. Increase i to $i+1$ and iterate from step 1 until a maximum number of iterations is reached.

For each speaker s and phone p, the final iteration produces phone-normalised acoustic features $\tilde{\boldsymbol{O}}_{s,p}$, cMLLR phone transforms $\tilde{\boldsymbol{W}}_p$ and phone-normalised speaker models $\tilde{\boldsymbol{\Lambda}}$.

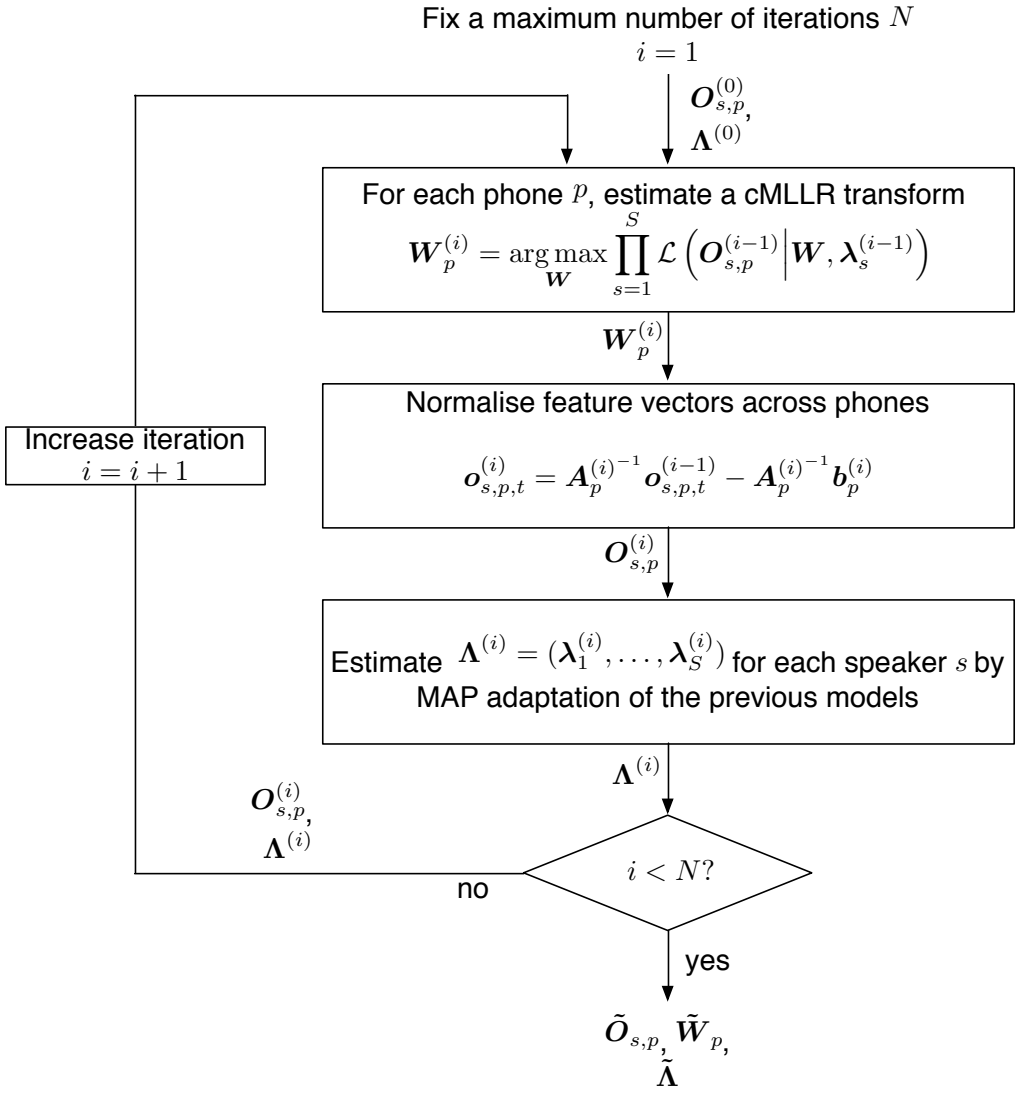


Fig. 6.4 An illustration of the PAT algorithm.

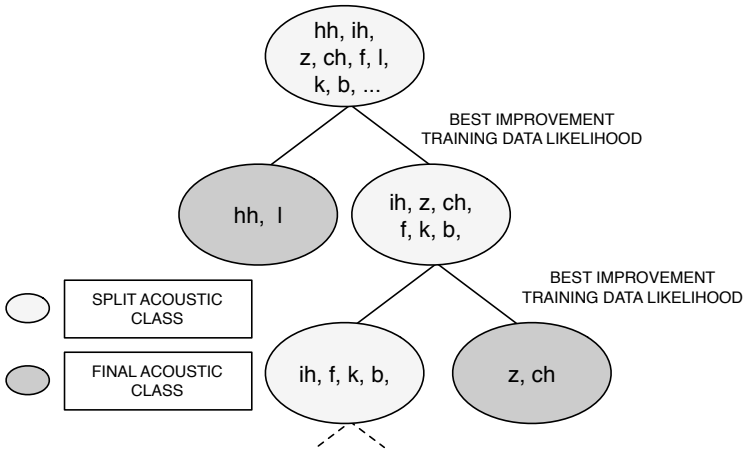


Fig. 6.5 An illustration of regression tree analysis which is used to identify suitable acoustic classes or groups of phones for PAT.

Acoustic classes

In practice, due to data limitations, it can be preferable to learn transforms $\tilde{\boldsymbol{W}}_p$ for groups of phones, often referred to as phone classes or acoustic classes, instead of individual phones. Based on linguistic analysis, suitable classes can be learned with a binary regression tree. As illustrated in Figure 6.5, the root node is initialized with a single acoustic class containing the full set of phones illustrated in Table 3.3. Each node is progressively split into smaller sub-classes for which separate transforms $\tilde{\boldsymbol{W}}_p$ are determined. The split is made according to that which maximises the data likelihood in Equation 6.13. The pooling of data according to acoustic classes, instead of phones, allows a more reliable estimation of a smaller set of transforms with less data.

PAT thus results in phone-normalised acoustic features from which more discriminant speaker models can be learned. In the next section, PAT performance is assessed through oracle ASV experiments performed on the TIMIT database [62] which is manually labelled at the phone level.

6.3 Oracle ASV experiments

This section reports experimental setup and results which analyse the performance of PAT under strictly controlled conditions by means of oracle ASV experiments. As illustrated in Figure 6.6, PAT is applied to the original acoustic features $\boldsymbol{O}_{s,p}$ according to the available ground-truth phonetic transcriptions. The transformed features $\tilde{\boldsymbol{O}}_{s,p}$

are then used for ASV experiments while the original acoustic features $\boldsymbol{O}_{s,p}$ for baseline ASV experiments. The difference between the obtained equal error rates (EERs) from the baseline ASV experiments and from the ASV experiments with PAT is considered as a measure of the speaker discriminative power of PAT in the case when speaker models are enrolled with short-duration utterances and in optimal conditions.

As already previously mentioned, in contrast to previous work [4] which was performed on the NIST Rich Transcription datasets in the context of speaker diarization, the work reported in this chapter has been performed on the TIMIT database (Chapter 3, Section 3.5) and in the context of ASV.

6.3.1 PAT performance

PAT performance is investigated using speaker models of between 4 and 1024 GMM components. Models are derived from the UBM, trained on the TIMITubm dataset, using conventional maximum a posteriori (MAP) adaptation. By means of the available ground-truth transcriptions, PAT transforms $\tilde{\boldsymbol{W}}_p$ for each phone $p \in P$ are then learned, as explained in Section 6.2.4 from a set of acoustic classes from the initial set of 38 phones illustrated in Table 3.3. A number of acoustic classes is controlled in the conventional manner with a regression tree and by fixing an initial desired likelihood. Independent transforms are learned for male and female speakers and for the set of utterances from the TIMITubm, TIMITspk and TIMITtest datasets.

The global PAT process (steps from 1 to 4) described in Section 6.2.4 was implemented with the Hidden Markov Model Toolkit (HTK) [86], in particular for creating the binary regression tree and for estimating the cMLLR transforms by solving Equation 6.15.

6.3.2 Speaker verification systems

PAT performance is assessed on two different ASV systems: a traditional GMM-UBM system and a state-of the art iVector-PLDA system. Baseline experiments were performed using the initial set of features $\boldsymbol{O}_{s,p}$ (or derived iVectors) used in PAT initialisation while ASV experiments with PAT are performed using the phone-normalised speaker features $\tilde{\boldsymbol{O}}_{s,p}$ (or derived iVectors) previously defined in section 6.2.4. For the iVector-PLDA system, the total variability matrix was estimated using the data from the TIMITubm dataset. Due to data limitations and since the aim is

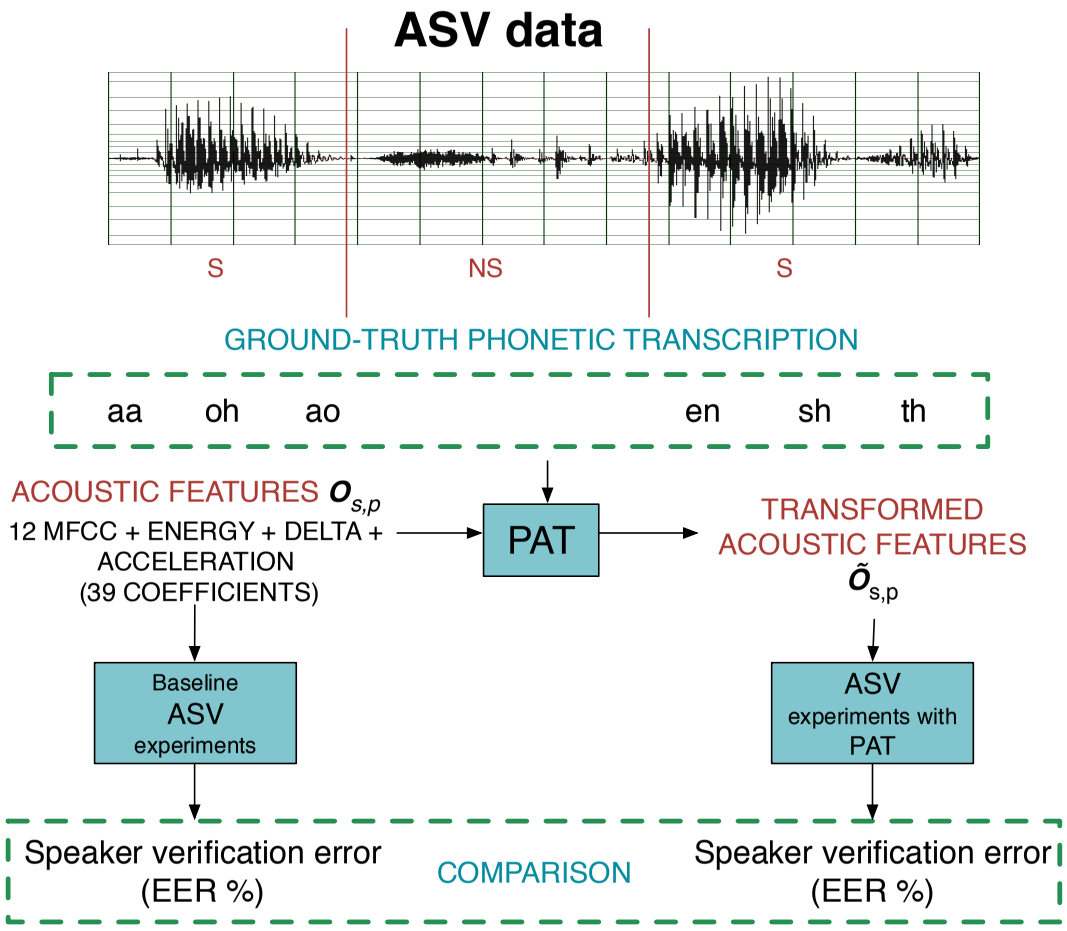


Fig. 6.6 An illustration of the experimental setup of the oracle ASV experiments.

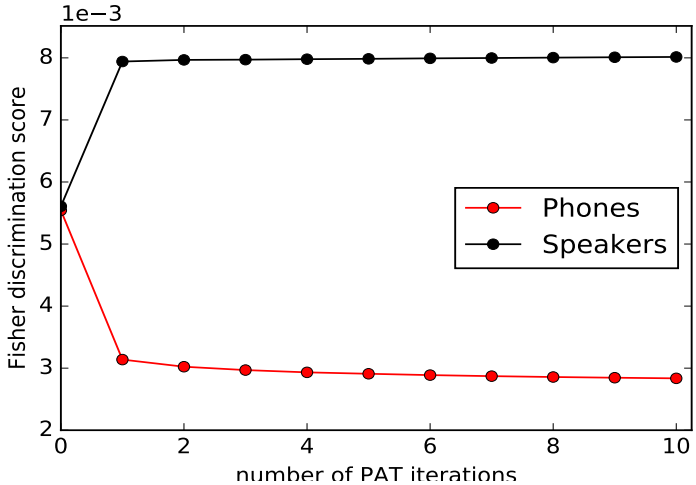


Fig. 6.7 Average phone and speaker discrimination for up to 10 iterations of PAT. Results shown for the 112 male speakers in the test dataset.

not to optimise ASV, but to observe the difference in ASV performance with PAT, the PLDA model is learned with the same development iVectors.

6.3.3 Experimental Results

PAT performance is analysed first, in terms of speaker and phone discrimination statistics, and second, in terms of its impact on ASV performance.

Speaker and phone discrimination

As reported previously in [4, 60], speaker and phone discrimination can be assessed at the feature level in terms of Fisher scores. They reflect the ratio of inter and intra class variance, where classes infer the subset of features corresponding to distinct speakers or distinct phones.

Given C_i, $i = 1, \ldots, S$ classes (phones or speakers) and a set of N labelled features $\boldsymbol{o}_t$, $t = 1, \ldots, N$ with $T_i = \{t|\boldsymbol{o}_t \in C_i\}$, the Fisher score is defined as follows:

$$S_{Fisher} = \frac{\sum_{i=1}^{S}\sum_{j=1}^{S}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)}{\sum_{l=1}^{S}\sum_{t\in T_l}(\boldsymbol{o}_t - \boldsymbol{\mu}_l)^T(\boldsymbol{o}_t - \boldsymbol{\mu}_l)} \tag{6.17}$$

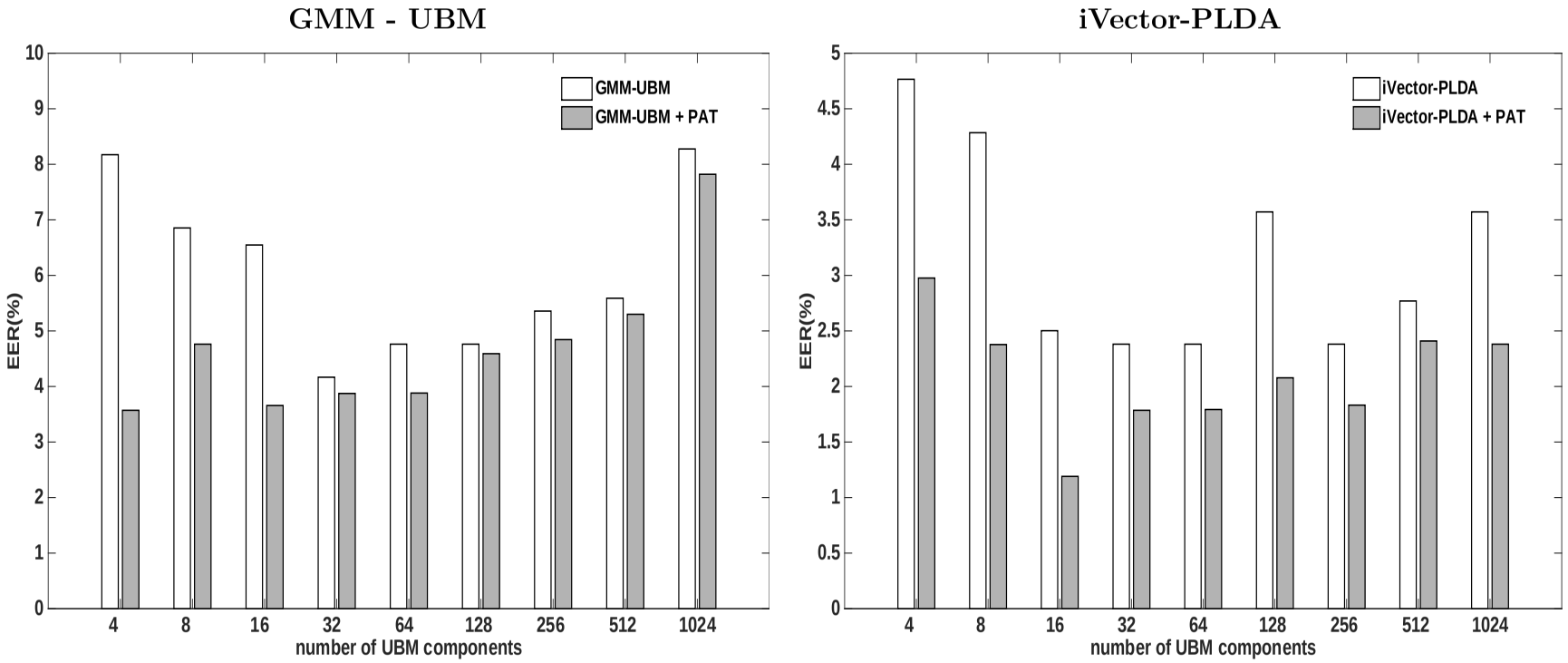
where $\boldsymbol{\mu}_i$ is the mean for class C_i and $\boldsymbol{o}_t$ is the t-th feature in the subset corresponding to class C_l.

Figure 6.7 illustrates average phone and speaker discrimination for the 112 male speakers in the test dataset. Discrimination is plotted as a function of PAT iterations. As expected, PAT reduces the phone discrimination (dashed profile) significantly. A rapid drop in phone discrimination occurs after a single iteration, probably due to the use of acoustic classes, tying together different phonemes and estimating the same cMLLR transform for the phones belonging to the same acoustic class. The algorithm converges with 10 iterations, after which the phone discrimination is approximately 50% lower than without PAT. Importantly, PAT also enhances speaker discrimination (solid profile). Figure 6.7 shows that after 10 iterations, speaker discrimination increases by approximately 43%. Features exhibiting lower phone discrimination but higher speaker discrimination should result in more discriminative speaker models. While improvements in ASV performance might be modest when training data is plentiful (models will be inherently phone-normalised without PAT), performance should improve in the case of limited training data. The ASV experiments presented in the next subsection seek to verify this hypothesis.

Automatic Speaker verification

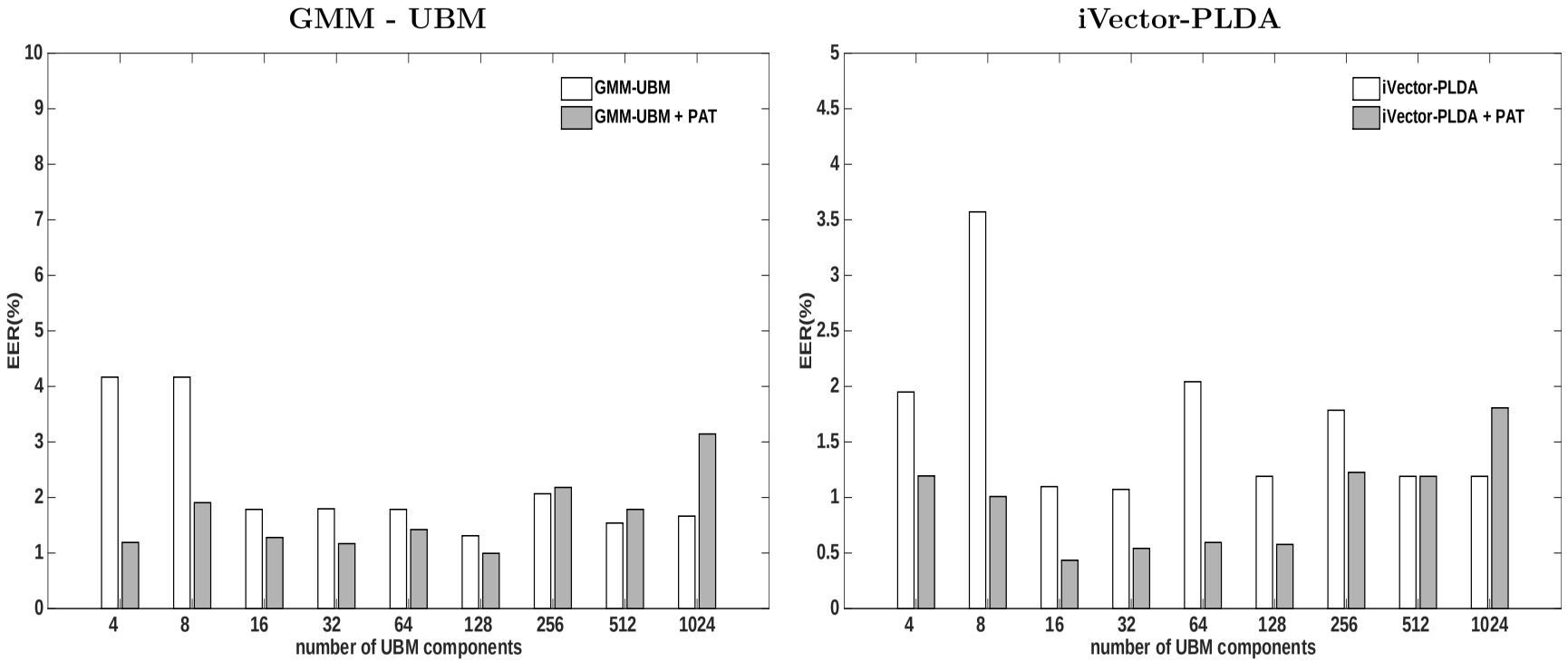
Figures 6.8, 6.9, 6.10 and 6.11 illustrate the performance in terms of equal error rate (EER) of GMM-UBM (left bar plots) and iVector-PLDA (right bar plots) systems, with and without PAT, for model sizes between 4 and 1024 components and for models trained with 1, 3, 5 or 7 TIMIT sentences respectively. In all cases, baseline performance is illustrated with clear bars. In general, as the amount of training data increases, then better performance is obtained with increasingly complex models. Noting the difference in scale between plots for each system, we also see that the iVector-PLDA system outperforms the GMM-UBM system when models are trained with relatively little data, whereas similar levels of performance are achieved when larger quantities are used.

We now turn to the assessment of PAT performance illustrated in Figures 6.8, 6.9, 6.10 and 6.11 by shaded bars. In general, for smaller model sizes and for both GMM-UBM and iVector-PLDA systems, performance with PAT is better than without – shaded bars are lower than clear bars. While improvements are mostly greater in the case



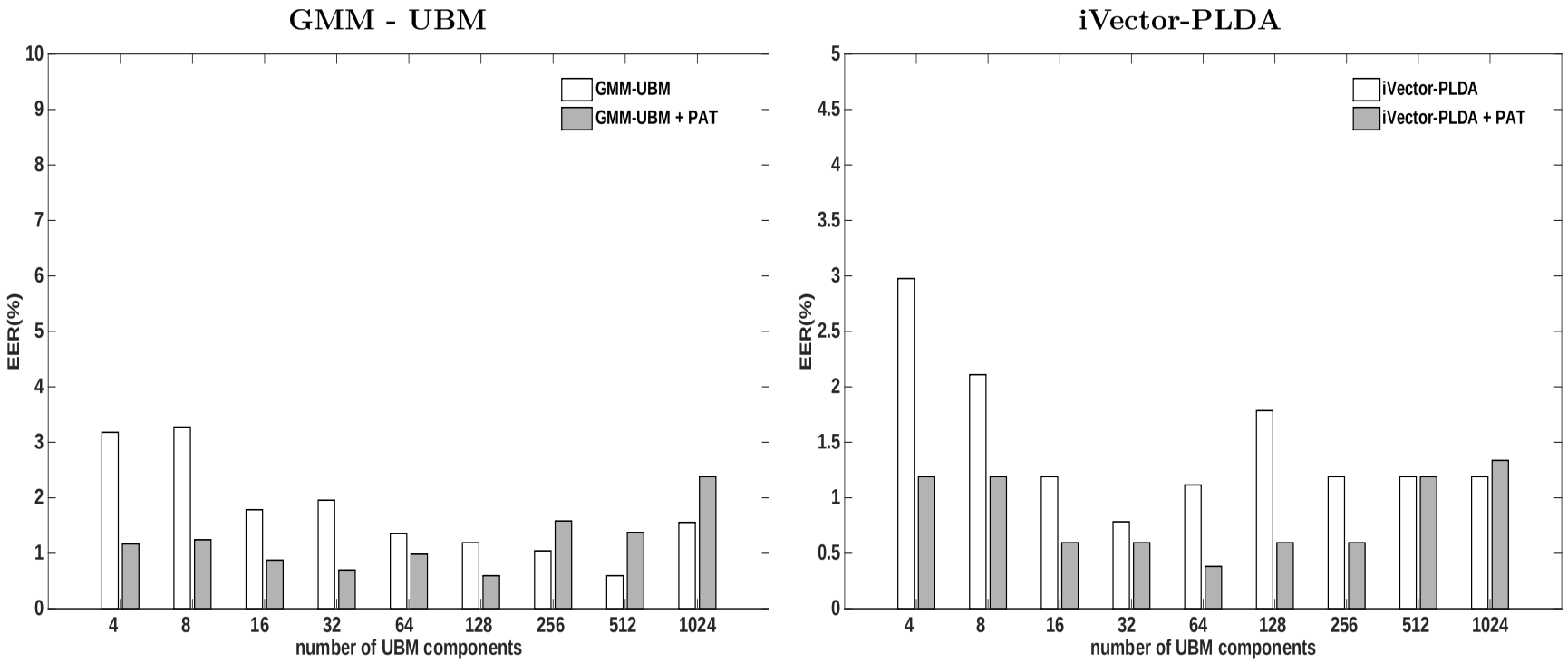
(a) models trained with 1 sentence

Fig. 6.8 An illustration of ASV performance for different model complexities (4-1024) and 1 TIMIT sentence to train speaker models. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars).



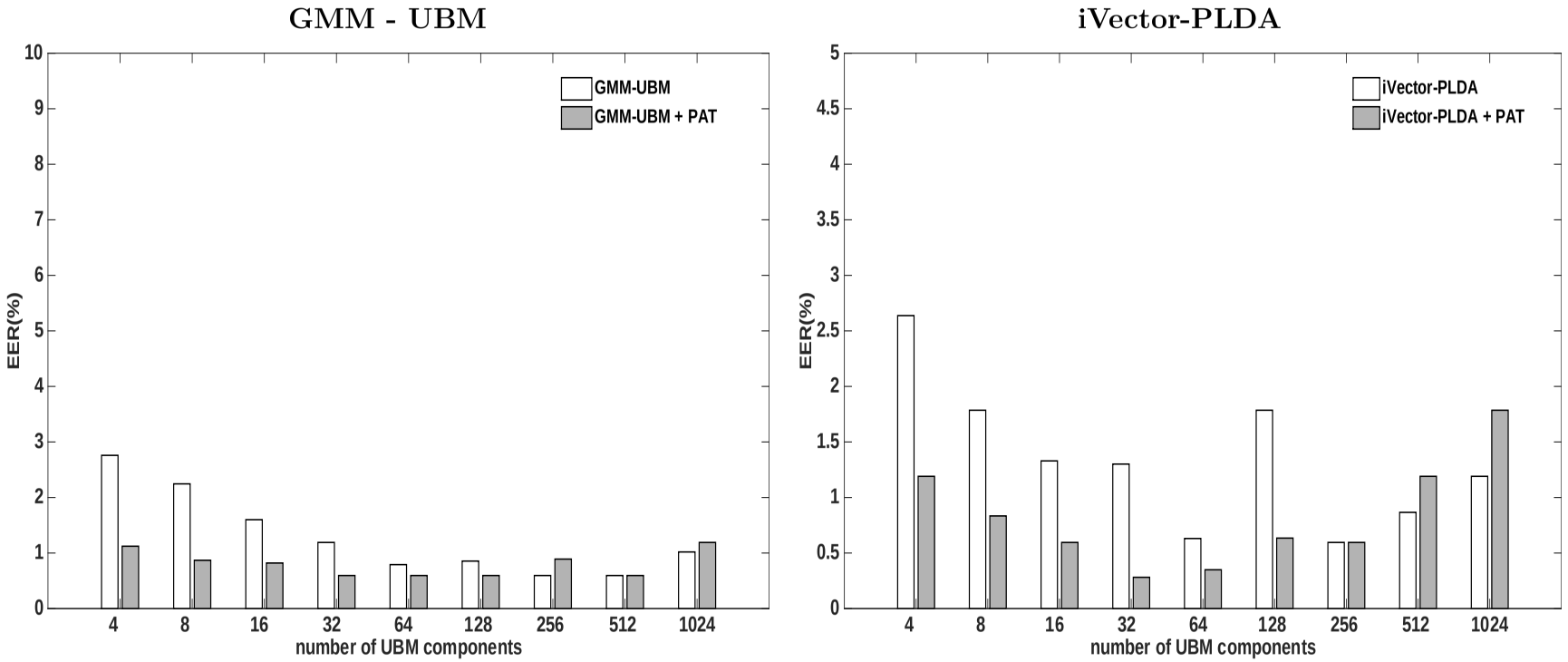
(a) models trained with 3 sentences

Fig. 6.9 An illustration of ASV performance for different model complexities (4-1024) and 3 TIMIT sentences to train speaker models. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars).



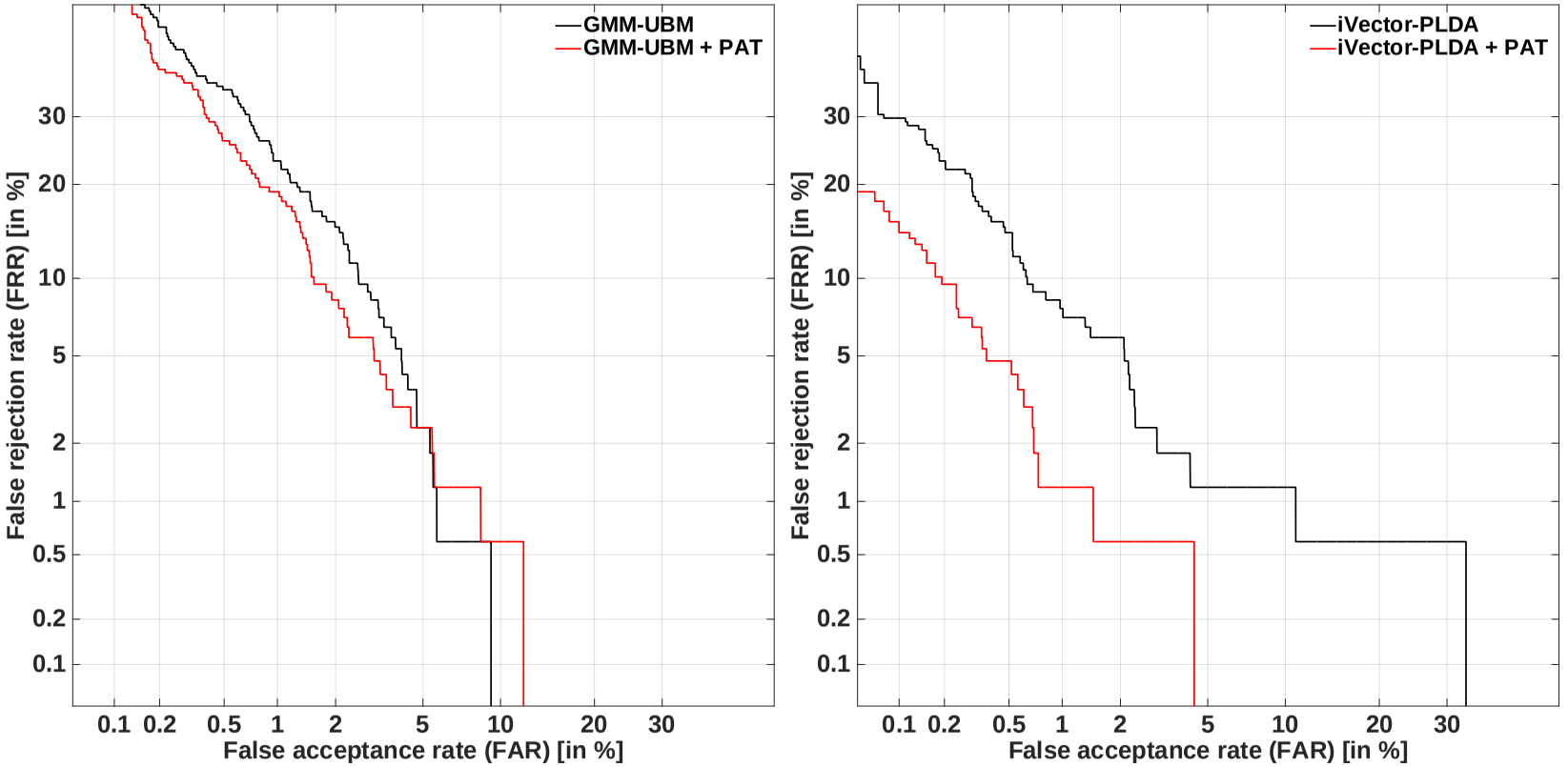
(a) models trained with 5 sentences

Fig. 6.10 An illustration of ASV performance for different model complexities (4-1024) and 5 TIMIT sentences to train speaker models. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars).



(a) models trained with 7 sentences

Fig. 6.11 An illustration of ASV performance for different model complexities (4-1024) and 7 TIMIT sentences to train speaker models. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars).



(a) GMM-UBM

(b) iVector-PLDA

Fig. 6.12 Detection error trade-off (DET) plots for GMM-UBM and iVector-PLDA systems with and without 5 iterations of PAT and for models trained with a single TIMIT sentence.

(a) GMM-UBM

Number of sentences for speaker model training	Baseline (EER %)	Baseline + PAT (EER %)
1	4.2	3.6
3	1.8	1.0
5	0.6	0.6
7	0.6	0.6

(b) iVector-PLDA

Number of sentences for speaker model training	Baseline (EER %)	Baseline + PAT (EER %)
1	2.4	1.2
3	1.1	0.4
5	1.1	0.4
7	0.6	0.3

Table 6.1 An illustration of EERs for the GMM-UBM and the iVector-PLDA systems with varying quantities of training data. Results shown for optimal model sizes in each case.

of low quantities of training data, modest improvements are also observed for the greatest quantities of training data. In some cases, for higher model sizes, PAT degrades performance. While it is difficult to explain these observations precisely, we expect this behaviour to be the result of over-fitting; with PAT, features are phone-normalised and accordingly require models of less complexity. Indeed optimal baseline performance is generally obtained with models of greater complexity than obtained by the same system with PAT.

Detection error trade-off (DET) profiles for both (a) GMM-UBM and (b) iVector-PLDA systems are illustrated in Figure 6.12. The two plots illustrate performance when speaker models of optimal size in each case are learned with only a single sentence (and thus corresponds to Figure 6.8a), with or without PAT. Baseline EERs of 4.2% and 2.4% are shown to fall to 3.6% and 1.2% with the application of PAT. PAT thus

delivers significant improvements in ASV performance in the case of short-duration training.

Table 6.1 illustrates a summary of performance for both GMM-UBM and iVector-PLDA systems for different quantities of training data. Results correspond to optimal model sizes in each case. When speaker models are trained on a single sentence, the baseline iVector-PLDA system outperforms the baseline GMM-UBM system by 43% relative (EERs of 4.2% c.f. 2.4%). When 7 sentences are used, both systems attain the same baseline EER of 0.6%. PAT leads to better or equivalent performance in all cases. When speaker models are learned with only a single sentence, baseline EERs decrease to 3.6% and 1.2% for the GMM-UBM and iVector-PLDA systems respectively. Of particular note, the greatest improvements in ASV performance are obtained for the iVector-PLDA system where performance is improved by 50% relative, irrespective of the quantity of training data.

6.4 Towards unsupervised PAT

The oracle experiments reported in Section 6.3 has sought to assess the performance of PAT under strictly controlled conditions during which PAT transforms are estimated according to the available ground-truth phonetic transcriptions in a completely supervised manner. However, in real scenarios, reliable phonetic transcriptions are rarely available and difficult to obtain. In this section, we thus present our efforts to develop PAT into a fully unsupervised system. Automatic acoustic class transcription is performed by means of an acoustic class recognizer whose output is used to estimate PAT transforms.

As illustrated in Figure 6.13, the data aimed to the estimation of the UBM from the TIMITubm dataset is used to determine from 5 to 38 acoustic classes by means of binary regression tree analysis. For each of the acoustic classes an HMM model is trained. These models are then fed into an automatic acoustic class recognizer in order to obtain acoustic class transcriptions. PAT is then applied on the original acoustic features $\boldsymbol{O}_{s,p}$ of the speech data according to the obtained acoustic class transcriptions rather than the original ground-truth phonetic transcriptions, as reported in Section 6.3. The transformed features $\tilde{\boldsymbol{O}}_{s,p}$ are then used for ASV experiments while the original acoustic features $\boldsymbol{O}_{s,p}$ for baseline ASV experiments. The difference between the obtained equal

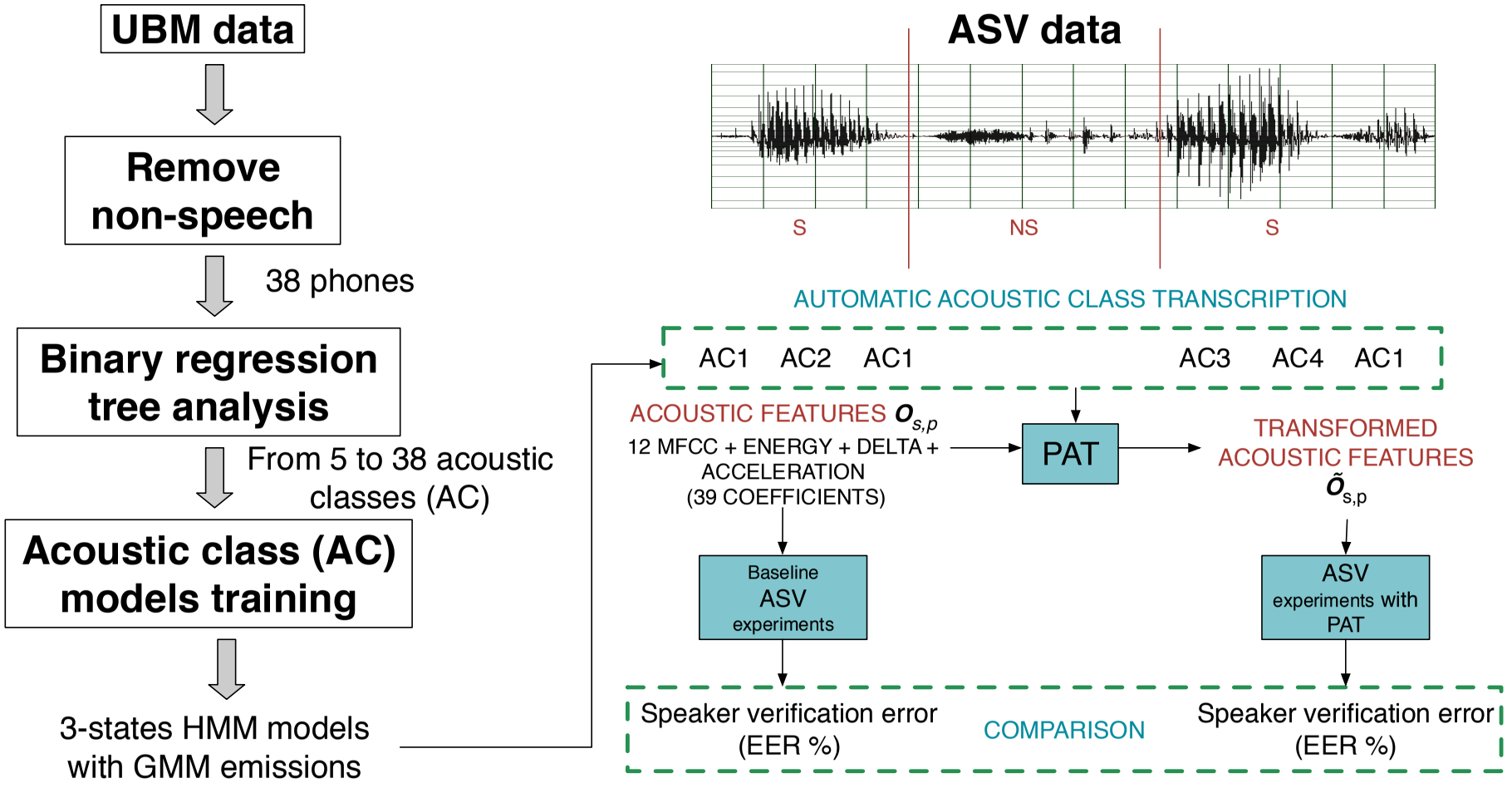


Fig. 6.13 An illustration of the experimental setup for unsupervised PAT.

error rates (EERs) of the baseline ASV experiments and of the ASV experiments with PAT is then considered as a measure to quantify the speaker discriminative power of PAT in the case when speaker models are enrolled with short-duration utterances and when automatic generated phonetic transcriptions are used.

6.4.1 Acoustic class transcription

An acoustic class recognizer is trained using the pool of acoustic features extracted from the TIMITubm dataset. By varying the likelihood threshold, the 38 phones in Table 3.3 (without silence) are reduced to between 5 and 38 acoustic classes through automatic binary regression tree analysis. For each number of acoustic classes, the phone labels in the phonetic transcriptions are replaced by their corresponding acoustic class labels.

The acoustic class models are 3-state hidden Markov models (HMMs) where each state is characterised by a Gaussian mixture model (GMM). Each acoustic class model is first initialized with a single Gaussian component whose mean and variance are set to that of the global class data. Subsequently, six iterations of embedded training are performed. The number of Gaussian components is doubled and embedded training is performed again on the new, larger model. This procedure is repeated until the number of Gaussian components reaches 128.

Subsequently, TIMITspk and TIMITtest datasets used for ASV experiments are transcribed automatically using the given set of acoustic classes and corresponding models previously trained. Both training and decoding phases were implemented with the Hidden Markov Model Toolkit (HTK) [86].

6.4.2 PAT and speaker verification

Analogous to oracle ASV experiments, PAT performance was investigated with automatically derived phone transcriptions using two different ASV systems: a traditional GMM-UBM system and a state-of the art iVector-PLDA system. Speaker models with between 4 and 1024 Gaussian components are derived from the UBM using conventional maximum a posteriori (MAP) adaptation. The features extracted from the TIMITubm dataset are treated with PAT which is applied using the TIMIT ground-truth transcriptions rather than the automatically derived transcriptions. All remaining data from the TIMITspk and TIMITtest datasets used for ASV experiments (model training

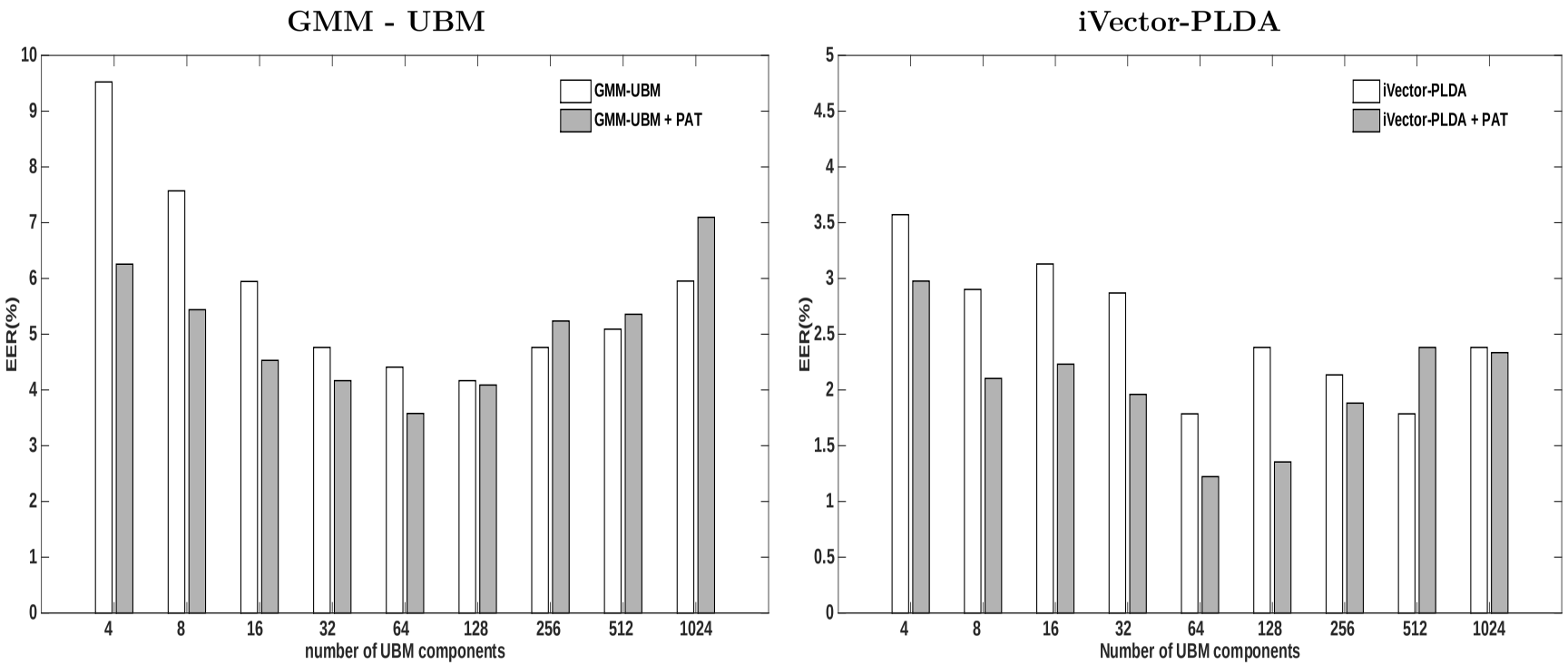
and testing) is instead treated with PAT applied using automatically derived acoustic class transcriptions as explained above. In both cases Equation 6.13 is applied with 5 iterations. As for acoustic class transcriptions, PAT was also implemented with the Hidden Markov Model Toolkit (HTK) [86]. Analogously to the oracle experiments, baseline ASV experiments were performed using the initial set of features $\boldsymbol{O}_{s,p}$ (or derived iVectors) while PAT performance was assessed using different numbers of acoustic classes and corresponding normalised features $\tilde{\boldsymbol{O}}_{s,p}$. As done for the oracle ASV experiments, for the iVector-PLDA system the total variability matrix was estimated on the TIMITubm dataset and the PLDA model is learned with the same development iVectors.

6.4.3 Experimental Results

Figures 6.14, 6.15, 6.16 and 6.17 illustrate the performance in terms of EER of GMM-UBM (left) and iVector-PLDA (right) systems, with and without PAT and using respectively 1, 3, 5 and 7 TIMIT sentences to train speaker models. Results are shown for model sizes between 4 and 1024 components and using 21 and 25 acoustic classes respectively. In all cases, baseline performance is illustrated with clear bars whereas that with 5 iterations of PAT is illustrated with shaded bars. In general, as the amount of training data increases, then better performance is obtained with increasingly complex models. The iVector-PLDA system outperforms the GMM-UBM system when models are trained with relatively little data, whereas similar level of performance are achieved when larger quantities are used.

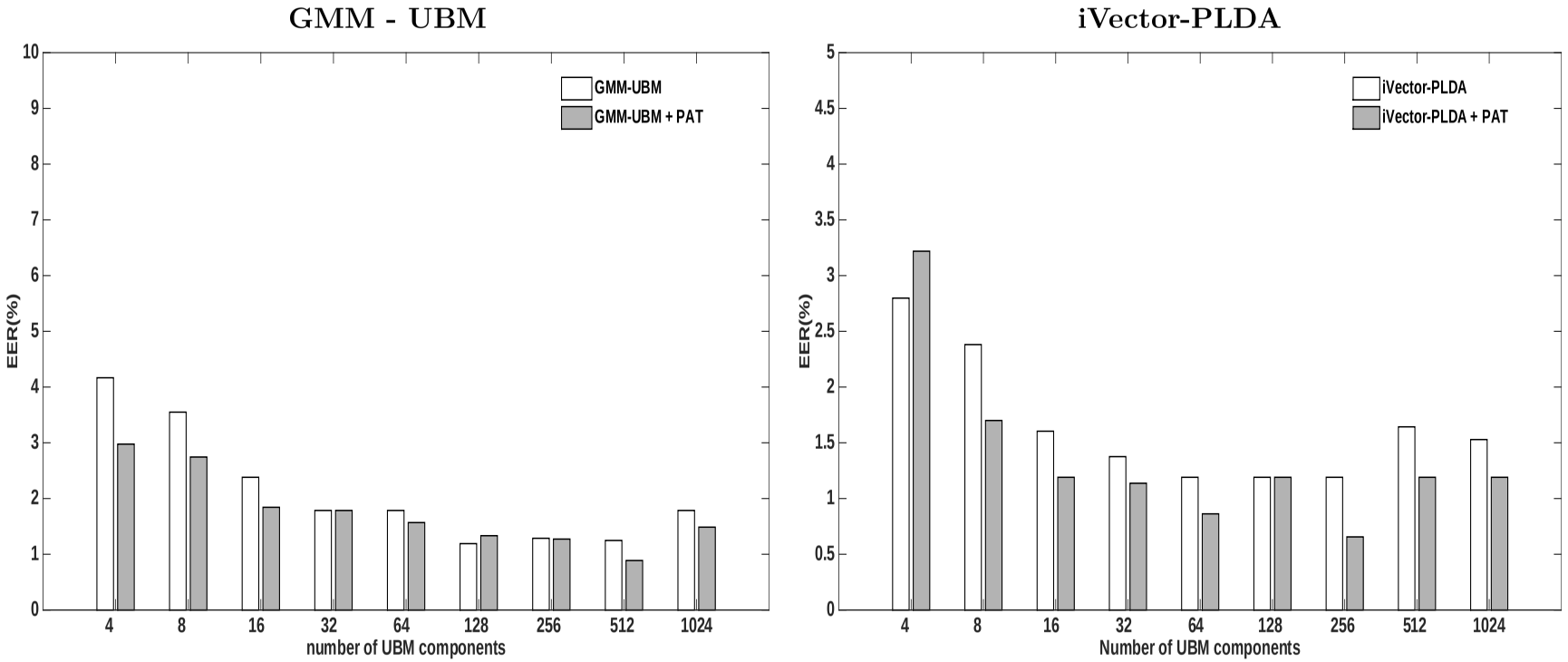
In general, for smaller model sizes and for both GMM-UBM and iVector-PLDA systems, performance with PAT is better than without – shaded bars are lower than clear bars. While improvements are mostly greater in the case of low quantities of training data, modest improvements are also observed for the greatest quantities of training data. With PAT, features are phone-normalised and accordingly require models of less complexity. Indeed, optimal baseline performance is generally obtained with models of greater complexity than obtained by the same system with PAT.

In Figure 6.14, for speaker models trained with a single sentence, the performance envelope for the GMM-UBM systems are convex with minima at 128 components for the baseline and 64 components with PAT. The iVector-PLDA profiles are somewhat



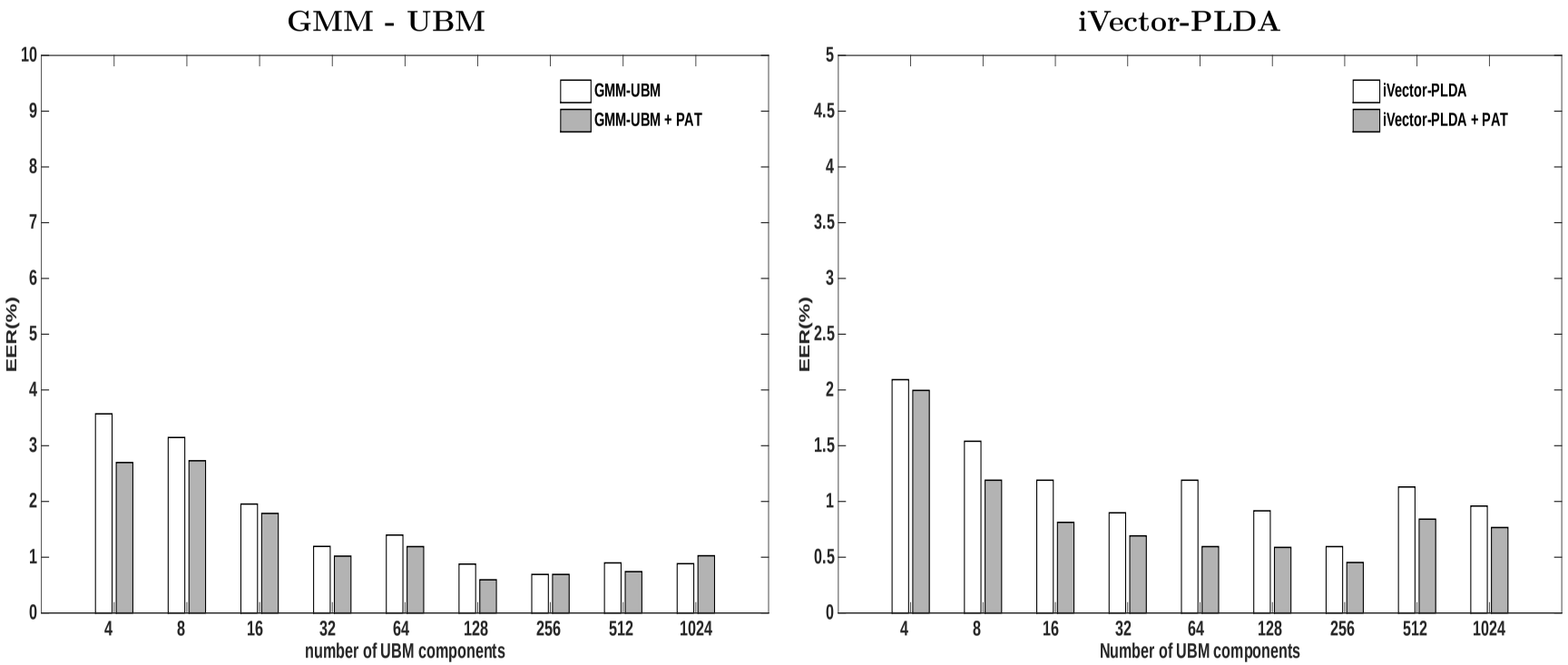
(a) models trained with 1 sentence

Fig. 6.14 An illustration of ASV performance for different model complexities (4-1024) and for speaker models trained with 1 TIMIT sentence. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars). PAT results are given for 21 acoustic classes in the case of GMM-UBM system and for 25 acoustic classes in the case of iVector-PLDA system.



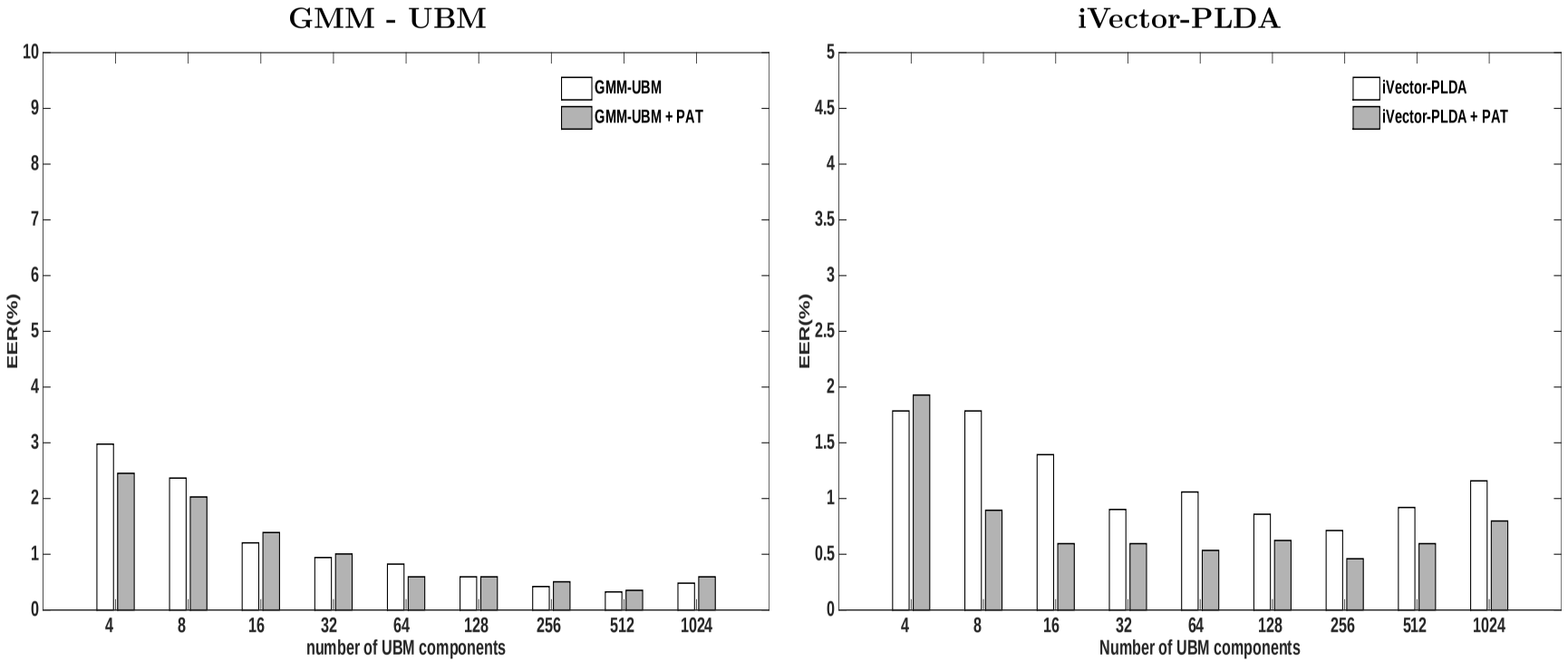
(a) models trained with 3 sentence

Fig. 6.15 An illustration of ASV performance for different model complexities (4-1024) and for speaker models trained with 3 TIMIT sentences. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars). PAT results are given for 21 acoustic classes in the case of GMM-UBM system and for 25 acoustic classes in the case of iVector-PLDA system.



(a) models trained with 5 sentence

Fig. 6.16 An illustration of ASV performance for different model complexities (4-1024) and for speaker models trained with 5 TIMIT sentences. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars). PAT results are given for 21 acoustic classes in the case of GMM-UBM system and for 25 acoustic classes in the case of iVector-PLDA system.



(a) models trained with 7 sentence

Fig. 6.17 An illustration of ASV performance for different model complexities (4-1024) and for speaker models trained with 7 TIMIT sentences. Plots show the EER for GMM-UBM (left) and iVector-PLDA (right) systems with (shaded bars) and without 5 iterations of PAT (clear bars). PAT results are given for 21 acoustic classes in the case of GMM-UBM system and for 25 acoustic classes in the case of iVector-PLDA system.

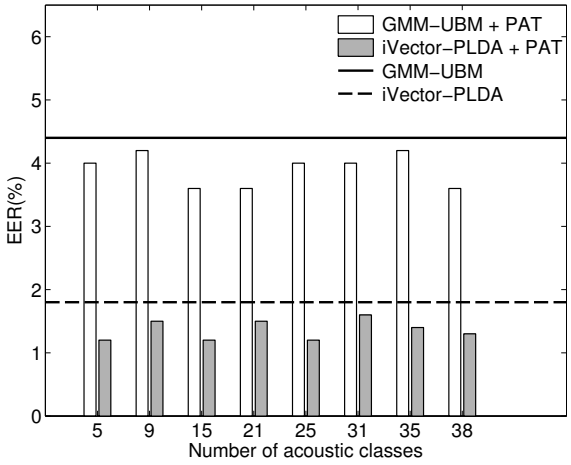


Fig. 6.18 An illustration of ASV performance for GMM-UBM and iVector-PLDA systems with 5 iterations of PAT for different numbers of acoustic classes, all for training data of 1 TIMIT sentence and for 64 UBM components. The baseline performance for GMM-UBM and iVector-PLDA systems are represented respectively by the solid and dashed horizontal lines.

noisy, mostly likely due to the lack of sufficient data to train the total variability matrix. Optimal performance with PAT is achieved for a model with 64 components.

Figure 6.18 illustrates PAT performance for the GMM-UBM system (clear bars) and the iVector-PLDA systems (shaded bars) with different numbers of acoustic classes. The complexity of both systems is fixed to 64 components. While the profile envelopes are non-convex, most likely again due to lack of training data, the application of PAT results in better performance than the respective baselines (solid and dashed horizontal lines). These observations indicate that PAT is beneficial even without reliable phone transcriptions. With 15 and 25 acoustic classes respectively the relative improvement in performance is 18% for the GMM-UBM system and 33% for the iVector-PLDA system.

Detection error trade-off (DET) profiles for the GMM-UBM and the iVector-PLDA systems using respectively 21 and 25 acoustic classes are illustrated in Figure 6.19. The profiles illustrate performance for optimal speaker model sizes trained using only a single TIMIT sentence, with and without PAT. The baseline EER of 4.2% and 1.8% are shown to fall respectively to 3.6% and 1.2% with the application of PAT. PAT thus delivers improvements in the case of speaker modelling with short-duration training utterances and even without accurate phonetic transcriptions.

(a) GMM-UBM

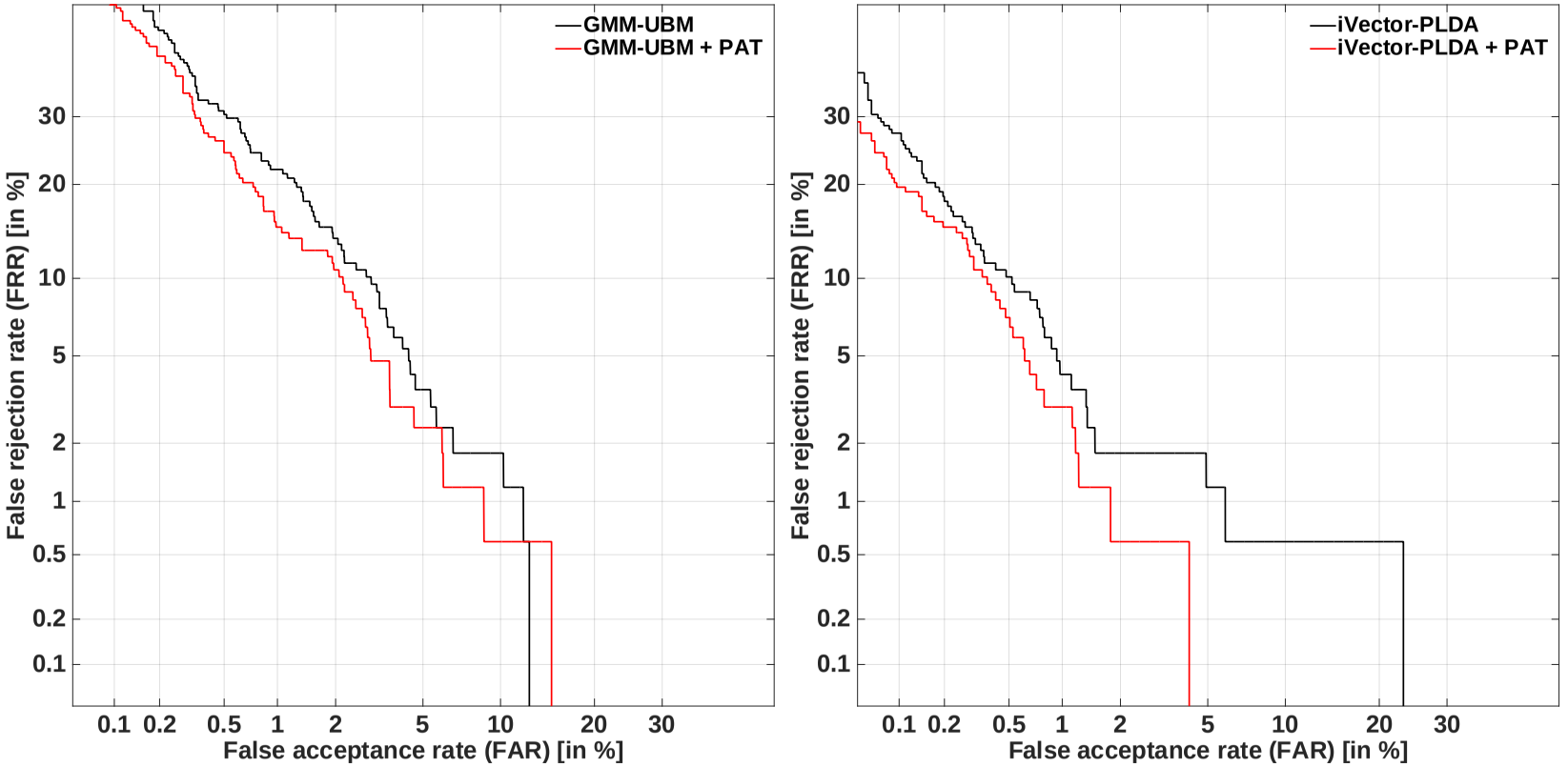
Number of sentences for speaker model training	Baseline (EER %)	Baseline + PAT (EER %)
1	4.2	3.6
3	1.2	0.9
5	0.7	0.6
7	0.3	0.4

(b) iVector-PLDA

Number of sentences for speaker model training	Baseline (EER %)	Baseline + PAT (EER %)
1	1.8	1.2
3	1.2	0.7
5	0.6	0.5
7	0.7	0.5

Table 6.2 An illustration of EERs for the GMM-UBM and the iVector-PLDA systems with varying quantities of training data. Results shown for optimal model sizes in each case. PAT results are given for 21 acoustic classes in the case of GMM-UBM system and for 25 acoustic classes in the case of iVector-PLDA system.

Table 6.2 illustrates a summary of performance for both the GMM-UBM and the iVector-PLDA systems for optimal model sizes and for different quantities of training data, namely 1 to 7 TIMIT sentences. PAT results (second column) are given for 21 acoustic classes in the case of GMM-UBM system and for 25 acoustic classes in the case of iVector-PLDA system. It is observed that, as the quantity of training data increases, then the difference between baseline and PAT performance decreases. This is to be expected since larger quantities of training data will inherently reduce the phone bias and have the same normalising effect as PAT. PAT thus delivers the most significant improvements in ASV performance in the case of short-duration training where the phone bias is otherwise the most pronounced.



(a) GMM-UBM

(b) iVector-PLDA

Fig. 6.19 Detection error trade-off (DET) plots for GMM-UBM and iVector-PLDA systems using 21 and 25 acoustic classes respectively, with and without 5 iterations of PAT and for models trained with a single TIMIT sentence.

6.5 Summary

Many automatic speech processing applications involving speaker modelling, such as text-independent automatic speaker verification (ASV) and especially on-line speaker diarization, are required to operate in the face of varying data quantities between training and testing. When training data is scarce, results might be biased due to the mismatching phonetic content encountered in the test data. Phone adaptive training (PAT) is a recent phone normalisation technique first applied to the context of speaker diarization. PAT is based on the application of constrained maximum likelihood linear regression (cMLLR) which aims to reduce phone influence at the feature level, while simultaneously emphasising speaker discrimination.

The first contribution presented in this chapter is the optimisation and evaluation of PAT at the speaker modelling level, with small-scale ASV oracle experiments performed on the TIMIT database, using the available ground-truth phonetic transcriptions and under strictly controlled conditions. PAT is successful in reducing phone bias and it improves significantly the performance of both traditional GMM-UBM and iVector-PLDA ASV systems in the case of short-duration training. Also of appeal, PAT can typically achieve better performance with less complex models.

The second contribution of this chapter consists of our efforts to develop PAT into a completely unsupervised system by means of an automatic approach to acoustic class transcription using binary regression tree analysis. PAT does not necessarily require accurate phone-level transcriptions. Results using the same ASV systems, show that PAT improves on baseline performance for all experiments with different numbers of acoustic classes and model complexities. Of particular note, the number of acoustic classes can be reduced significantly meaning that PAT is effective even without reliable phone transcriptions. This may ease the application of PAT to more realistic, noisy data where the estimation of phone transcription can be troublesome. Finally, improvements in performance tend to reduce as the amount of training data increases, meaning that PAT is most beneficial when training data is scarce.

In Chapter 7, a first attempt of applying PAT to semi-supervised on-line diarization, in order to improve speaker modelling and therefore reduce the system latency, is described.

Chapter 7

PAT for on-line diarization: a first attempt

Chapter 4 has reported our first attempt to develop a completely un-supervised on-line diarization system aimed at supporting new emerging practical applications, due to the spread of IoT, connected smart objects and always listening sensors. Although, obtained results are in line with other work in literature, the obtained high diarization error rates highlight even more the challenge involved.

After identifying the main bottleneck in the unsupervised initialisation of speaker models with accumulated short-speech segments, a semi-supervised on-line speaker diarization system, in which speaker models are initialised off-line with short amounts of manually labelled training data, has been proposed in Chapter 5. Although, the manual initialisation of speaker models might represent an inconvenience, the huge improvement in performance probably justifies the additional effort and opens the potential for the application of either supervised or semi-supervised speaker discriminant feature transformations. Such transforms may offer an opportunity for improved performance, by reducing the need for seed data, latency, or both. One avenue through which this objective might be pursued involves PAT introduced and described in Chapter 6.

In this chapter, a first attempt to apply PAT to semi-supervised on-line speaker diarization and under strictly controlled conditions is presented. Due to the unavailability of suitable datasets for speaker diarization transcribed at the phonetic level, multi-speakers conversations are simulated by joining different TIMIT audio files. Obtained audio files are then processed with the semi-supervised on-line diarization

system, described in Chapter 5. PAT transforms, trained on external data, are applied to the original acoustic features in the speech segments in order to reduce the phonetic variation. Despite the use of simulated data, experimental results presented in this chapter highlight the potential of PAT to improve the performance of semi-supervised on-line speaker diarization, in particular by reducing the quantity of initial seeding data and the speaker model complexity.

The remainder of this Chapter is organised as follows. Section 7.1 describes how the simulated conversations are obtained. Section 7.2 describes the experimental setup. Section 7.3 reports experimental results. Finally, some conclusions are drawn in Section 7.4.

7.1 Simulated conversations using TIMIT dataset

Due to the lack of databases transcribed at the phonetic level suitable for the development of on-line diarization together with PAT optimisation, the TIMIT database is used to create simulated audio conversations by joining different audio files. For this purpose, 5 recordings for each speaker from the TIMITspk and TIMITtest datasets are set apart. Each simulated multi-speaker conversation involves from 4 to 9 speakers whose related recordings are shuffled, repeated and joined multiple times with no overlap, in order to reach an average duration from 15 to 30 minutes. All the phonetic ground-truth and speaker transcriptions are processed accordingly so that they are aligned to the newly created audio conversations. A total of 15 simulated conversations are obtained.

7.2 System setup

On-line diarization experiments are performed using the semi-supervised on-line diarization system with incremental MAP adaptation described in Chapter 5.

PAT transforms are estimated according to the ground-truth transcriptions using all remaining sentences for each speaker which are not involved in the creation of audio conversations. The normalised acoustic features are then used for the enrolment of phone-normalised speaker models with different amount T_{SPK} of training data during the initialisation phase of the semi-supervised diarization system. During the on-line

classification of speech segments of a maximum fixed duration T_S, acoustic features are phone-normalised by applying the estimated PAT transforms and classified against the corresponding phone-normalised speaker models. Incremental MAP adaptation is used for the update of speaker models.

Baseline experiments are carried out in the same way as described in Chapter 5 with the original acoustic features and without the application of PAT. Figure 7.1 represents the semi-supervised on-line diarization combined with PAT transforms.

7.3 Experimental results

Semi-supervised on-line diarization performance, both with and without PAT are reported in Figures 7.2, 7.3, 7.4, 7.5 and 7.6, for segment durations of 0.25, 0.5, 1, 2 and 3 seconds respectively. For the results shown in the left bar plots speaker models are initialised with an amount of training data T_{SPK} of 5 seconds while for the results shown in the right bar plots a duration T_{SPK} of 7 seconds is utilised. In all cases, baseline performance is illustrated with clear bars. Diarization performance with PAT is instead illustrated by shaded bars. In general, performance with PAT is better than without – shaded bars are lower than clear bars.

Analogous to the results reported in Chapter 6 for ASV experiments, it is observable that almost always the best baseline performance is surpassed with PAT when using a lower model complexity. For instance, in Figure 7.2 for a segment duration T_S of 0.25 seconds and a training duration T_{SPK} of 5 seconds (left plot), a baseline DER of 25% is obtained with a 64 Gaussian components model while by applying PAT a DER of 22% is reached with a 32 Gaussian components model. Similarly, in Figure 7.4 for a segment duration T_S of 1 second and a training duration T_{SPK} of 7 seconds (right plot), a baseline DER of 10.5% is reached with a 64 Gaussian components model while with PAT application a DER of 9.5% is reached with a 32 Gaussian components model.

Applying PAT with a lower amount of training data T_{SPK} can result in the same performance as the baseline. For instance in Figure 7.4, a DER of 10.5% is reached by the baseline with a training duration T_{SPK} of 7 seconds and with a 64 Gaussian components model. The same performance is reached by applying PAT with a training duration T_{SPK} of 5 seconds and with a 32 Gaussian components model.

Although, these experiments are based on simple, simulated conversation data, results show the potential of PAT in improving semi-supervised on-line diarization, by contributing to reduce the model complexity and in some cases the amount of required labelled training data. Despite this, the advancing in this direction is clearly limited by the lack of a phonetic labelled databases suitable for the development of on-line diarization together with the optimisation of PAT.

7.4 Summary

Since on-line diarization involves the continuous training, update and comparison of speaker models with short-duration speech segments, similarly to many other automatic speech processing applications, its performance is strongly affected by the variation due to the phonetic content. PAT is a technique to marginalise the phonetic variation while increasing the speaker discrimination, by projecting the original acoustic features into a more speaker discriminative space. In Chapter 6, it has been shown that PAT is able to improve short-duration speaker modelling in ASV both with a GMM-UBM system and a state-of-the-art iVector-PLDA system. The main scope of this Chapter is to highlight how the application of PAT transforms could be potentially beneficial for on-line diarization both to reduce the system latency and in the case of semi-supervised on-line diarization to reduce the amount of labelled training data required by the initialisation of speaker models. Even though the audio data used for experiments are simulated, experiments show that PAT could improve the performance of a semi-supervised on-line diarization system. Similar performance to a baseline system could be reached with lower complexity models and less amount of labelled training data. Despite the potential of PAT in improving short-duration speaker modelling, one of the main obstacle to further advance into this direction is the lack of manually phonetic transcribed databases suitable for diarization purposes together with PAT optimisation, therefore making the choice of TIMIT database mandatory.

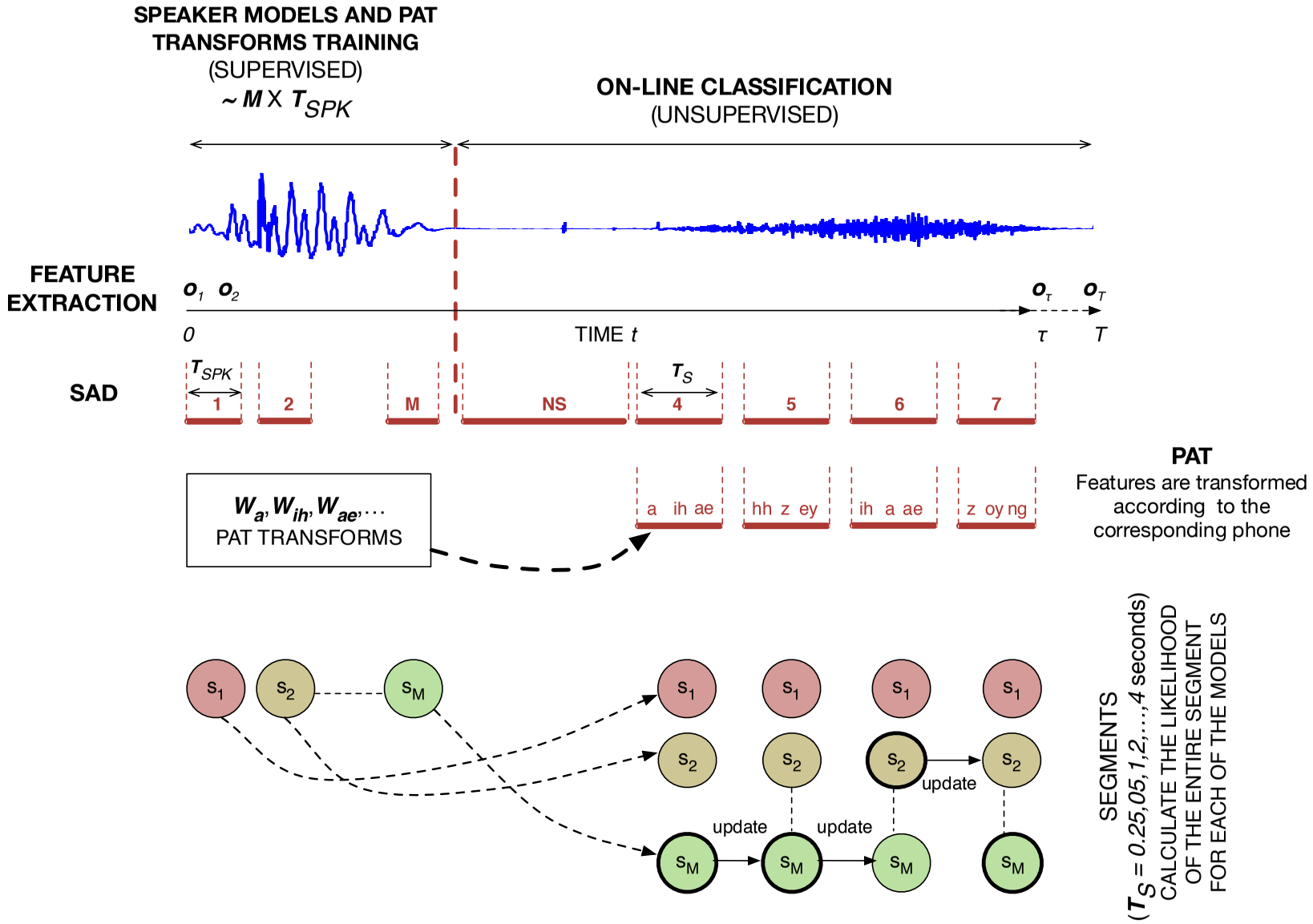
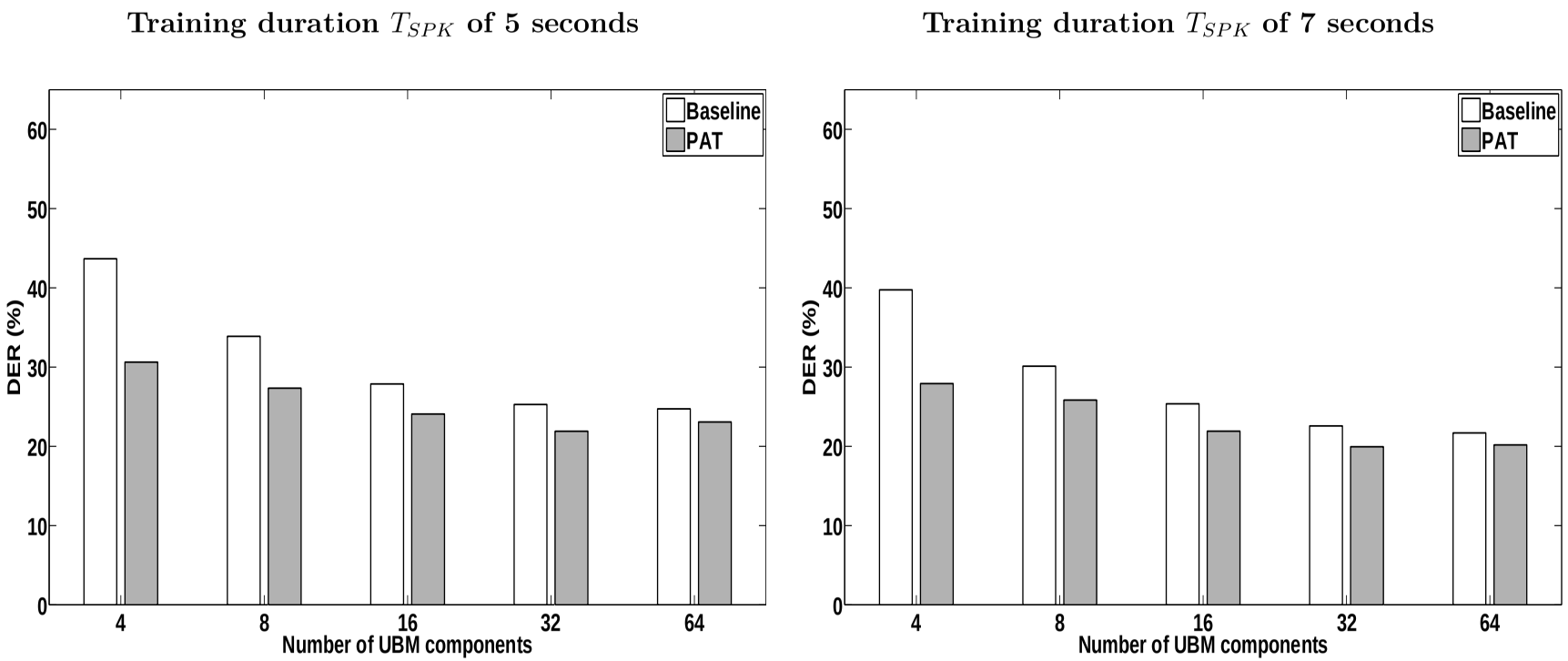
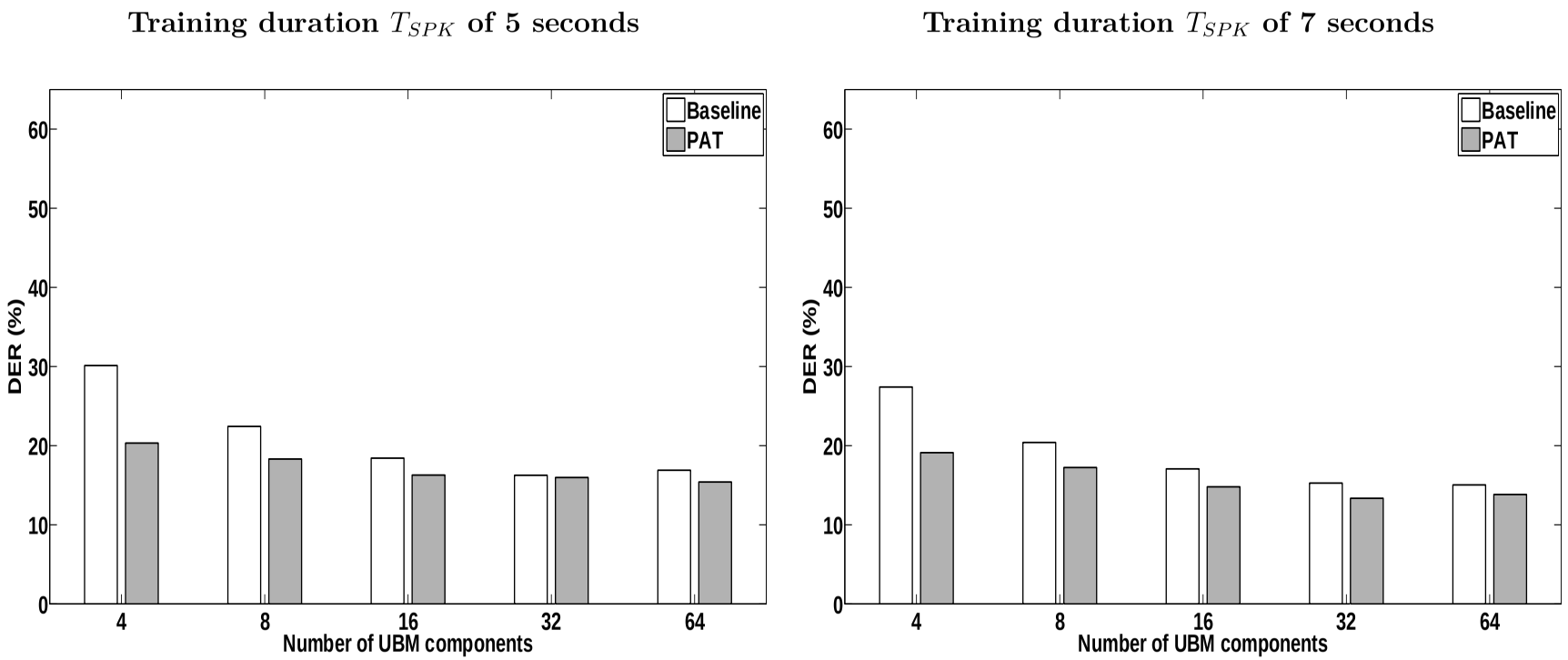


Fig. 7.1 An illustration of the implemented semi-supervised on-line diarization system with PAT application.



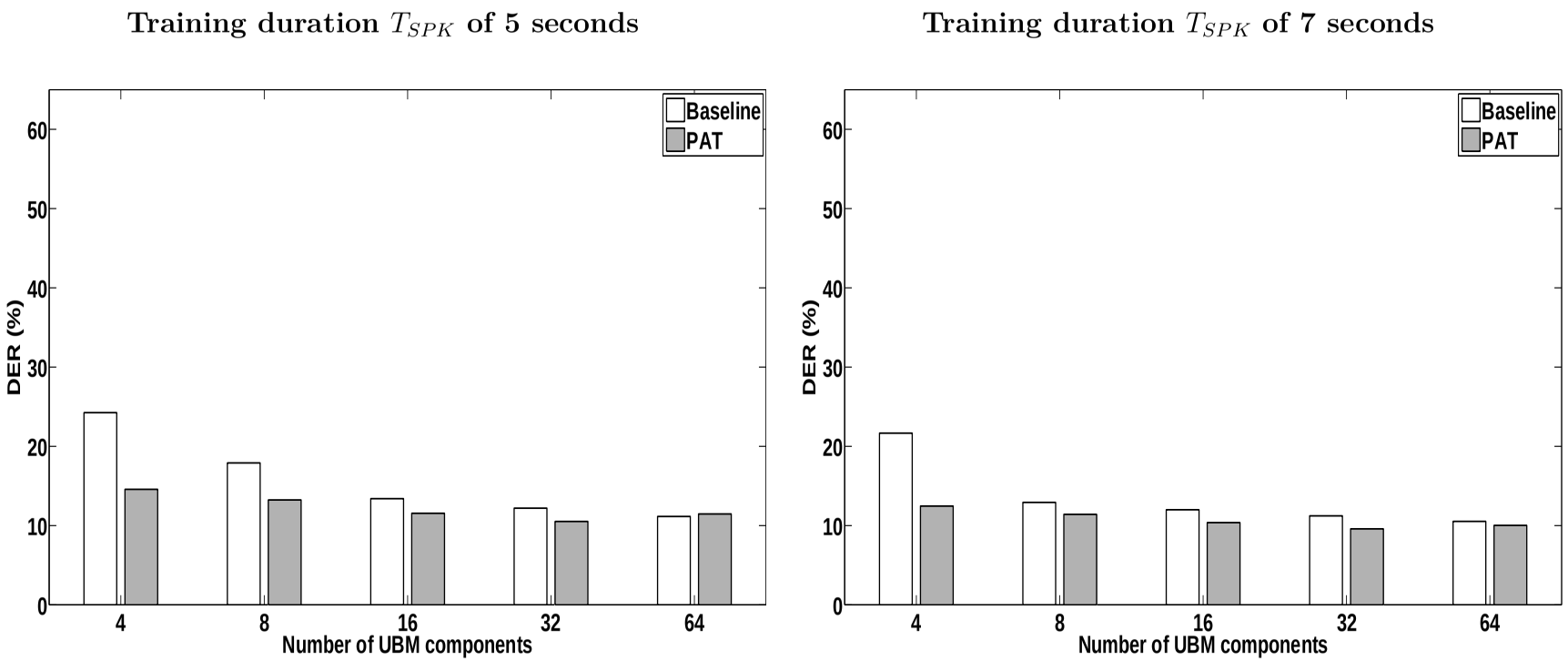
(a) Segment duration $T_S = 0.25$ seconds

Fig. 7.2 An illustration of semi-supervised on-line speaker diarization performance for different model complexities (4-64) and for a segment duration T_S of 0.25 seconds. Plots show the DER for 5 seconds (left) and 7 seconds (right) of training data with (shaded bars) and without 5 iterations of PAT (clear bars).



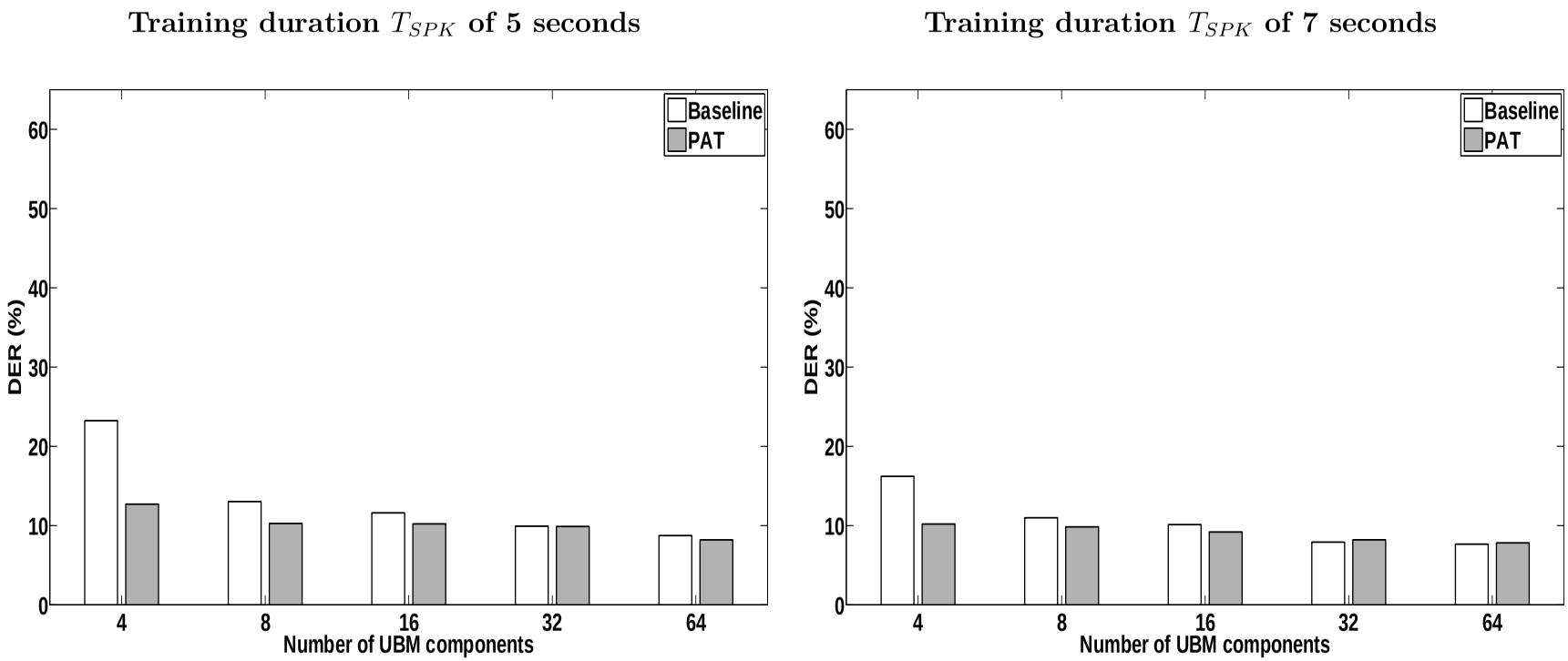
(a) Segment duration $T_S = 0.5$ seconds

Fig. 7.3 An illustration of semi-supervised on-line speaker diarization performance for different model complexities (4-64) and for a segment duration T_S of 0.5 seconds. Plots show the DER for 5 seconds (left) and 7 seconds (right) of training data with (shaded bars) and without 5 iterations of PAT (clear bars).



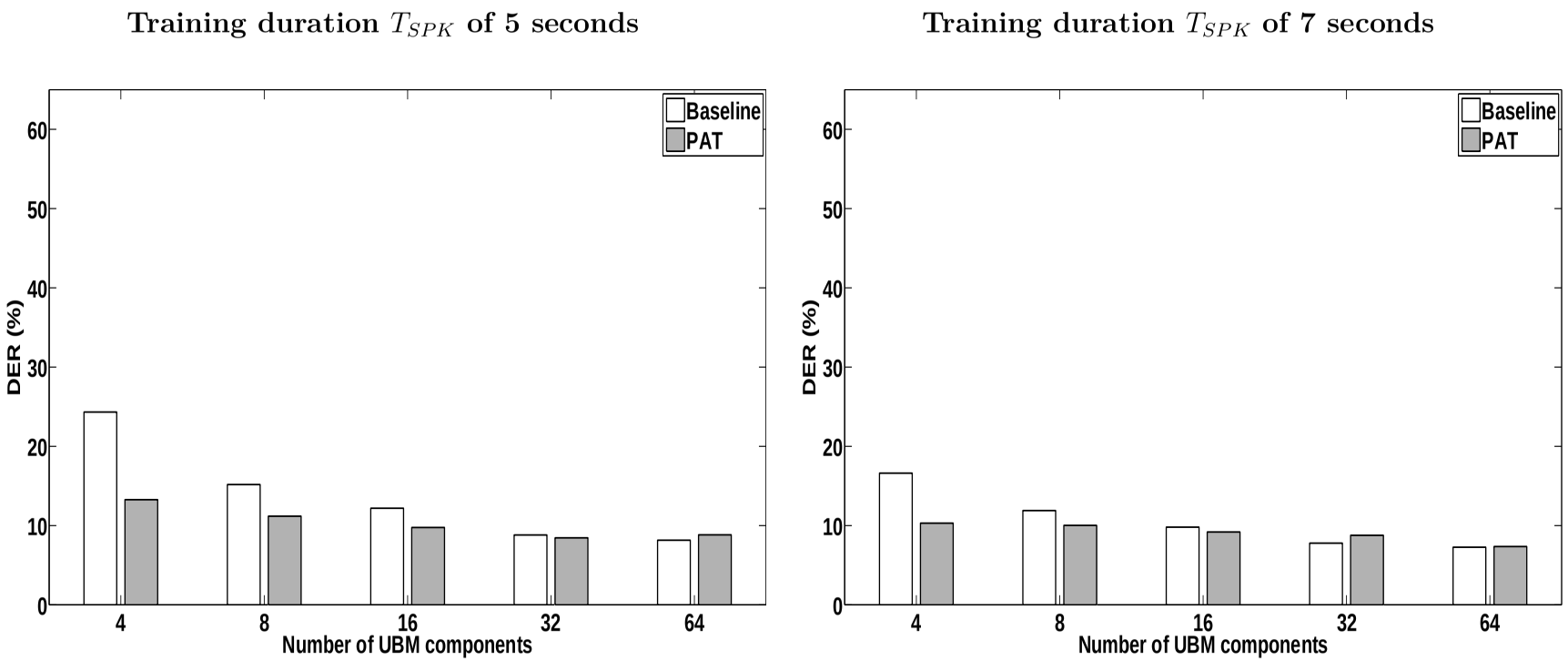
(a) Segment duration $T_S = 1$ second

Fig. 7.4 An illustration of semi-supervised on-line speaker diarization performance for different model complexities (4-64) and for a segment duration T_S of 1 second. Plots show the DER for 5 seconds (left) and 7 seconds (right) of training data with (shaded bars) and without 5 iterations of PAT (clear bars).



(a) Segment duration $T_S = 2$ seconds

Fig. 7.5 An illustration of semi-supervised on-line speaker diarization performance for different model complexities (4-64) and for a segment duration T_S of 2 seconds. Plots show the DER for 5 seconds (left) and 7 seconds (right) of training data with (shaded bars) and without 5 iterations of PAT (clear bars).



(a) Segment duration $T_S = 3$ seconds

Fig. 7.6 An illustration of semi-supervised on-line speaker diarization performance for different model complexities (4-64) and for a segment duration T_S of 3 seconds. Plots show the DER for 5 seconds (left) and 7 seconds (right) of training data with (shaded bars) and without 5 iterations of PAT (clear bars).

Chapter 8

Summary & conclusions

On-line diarization and its practical implementation has gained attention in recent years, mainly due to an increase in interest for the need of speech-based context awareness applications, based on the data collected from listening sensors in a multi-speaker environment.

Although few works in the literature have addressed the problem of on-line diarization, a common trait among them is that the diarization techniques are developed on less challenging broadcast news and plenary speeches recordings, characterized by longer speaker turns and low spontaneity. Moreover, the proposed solutions are not particularly suited to support emerging practical applications due to the high error diarization rates.

Thus, the focus of this dissertation has been on the development of practical and computationally efficient on-line diarization systems for more challenging recordings, for instance recordings from a meeting room. In addition, the problem of linguistic and phonetic variation, affecting short-duration automatic speaker verification (ASV) and on-line diarization systems, is also addressed. In this regard, Phone Adaptive Training (PAT), a recently proposed phone normalisation technique, is further optimised and developed towards a complete unsupervised system.

A summary of the contributions in each chapter is given in Section 8.1, while future works is introduced in Section 8.2.

8.1 Contributions

A list of the main contributions and results for each chapter of this thesis is provided below:

- Chapter 4 deals with the problem of on-line speaker diarization and it proposes a new adaptive, unsupervised on-line approach to speaker diarization for meeting data captured with a single distant microphone. The performed experiments show that the best performance implies a latency in the order of 3 or 4 seconds and the accuracy of the trained speaker models converges as the amount of training data increases. While results are in line with those reported for less challenging data, diarization error rates remain high, probably too high to support any practical applications. This is mainly due to the unsupervised on-line initialisation of speaker models and their subsequent adaptation with short-duration speech segments as shown in Chapter 5.

- Chapter 5 reports a semi-supervised on-line diarization system in which speaker models are seeded with an initial amount of labelled training data. Relaxing the supervision constraints allows the initialisation of reliable speaker models ready for on-line classification. The use of longer segments might contain multiple speakers and increase drastically the system latency. Such a system can outperform an off-line diarization system with just few seconds of speaker seed data and 3 seconds of latency when using an incremental MAP adaptation procedure in the case of the RT07 meetings dataset. By using greater quantities of seed data or by allowing greater latency a diarization error rate in the order of 10% can be achieved.

- Chapter 6 focuses on the development and optimisation of PAT, a phone normalisation technique based on cMLLR transform developed to marginalise the phonetic variation. The first contribution of this chapter is the evaluation and optimisation of PAT at the speaker modelling level, by means of small-scale oracle ASV oracle experiments on the TIMIT database, using the available ground-truth phonetic transcriptions and under strictly controlled conditions. Results reported in this chaptee show that PAT is successful in reducing phone bias and it improves significantly the performance of both traditional GMM-UBM and iVector-PLDA

ASV systems in the case of training and testing with short-duration sentences. The second contribution of this chapter is the development of PAT towards a completely un-supervised system to support potential practical applications. An automatic approach to acoustic class transcription using binary regression tree analysis is proposed. Experimental results using the same ASV systems and with different number of acoustic classes show that PAT is still beneficial even without reliable phonetic transcriptions. In all cases, with less complex models PAT can typically achieve better performance than the baseline.

- Chapter 7 presents our first attempt to combine PAT with semi-supervised on-line speaker diarization. As already mentioned, on-line speaker diarization requires the iterative learning and update of speaker models with short-duration speech segments. Due to the short amount of speech data, speaker models might be biased towards the phonetic content. Due to the lack of phonetic labelled databases suitable for the development of on-line diarization together with PAT, simulated audio recordings are created by joining different audio recordings. Even though experiments are performed on simulated conversations, experimental results show the potential of PAT in improving the performance of a semi-supervised on-line diarization system by using less complex speaker models and by requiring shorter amount of labelled training data for speaker models initialisation.

8.2 Future works

This dissertation highlights the challenges involved in the development of a usable on-line diarization system due to the initialisation of speaker models and their following comparison with short-duration speech segments. It also highlights the potential of PAT in marginalising the phonetic variation when speaker models are trained using short-duration utterances in both ASV and on-line diarization scenarios.

Further work is required to address the following:

- **On-line speaker change detection:** both the unsupervised and semi-supervised on-line diarization systems, reported in this thesis, rely on the uniform splitting of the speech segments according to a fixed maximum duration T_S without taking into consideration the actual speaker boundaries. Despite being computationally

efficient, this approach involves obviously the risk of including more than one speaker in the same speech segment, ultimately leading to the initialisation or adaptation of impure speaker models and accumulation of errors. Therefore, the development and application of on-line speaker change detection would allow to identify more precisely the speaker turns boundaries, thus providing purer speech segments which potentially contains a single speaker and ultimately better diarization results.

- **Hybrid diarization system:** un-supervised on-line diarization has to produce decisions on- the-fly that will inevitably degrade diarization performance below that achievable with a strictly off- line diarization system. Furthermore, a strictly on-line diarization system has no capacity to correct for earlier mistakes (for example re-segmentation or re-alignment) which implies the potential for their accumulation and, ultimately, unreliable diarization. While, these problems could be partially solved by relaxing the supervision constraint as shown in this thesis, the advanced knowledge of the speaker numbers and the initial seeding of speaker models might not be possible in all situations. A possible alternative could be the application of a low-resource, secondary off-line diarization process which runs in parallel to the main on-line approach. Its main aim is to identify and correct previous mistakes and therefore the accumulation of errors. Speaker models can then be re-learned or re-adapted with the benefit of more reliable decisions taken off-line. A suitable choice for the off-line diarization system could be the diarization system based on binary keys, described in Chapter 2, Section 2.6.4 thanks to its important computational efficiency.

- **PAT transforms adaptation:** even though PAT transforms could be estimated for more general acoustic classes rather than for single phones as shown in Chapter 6, the estimation of PAT still requires a large amount of training data. Moreover, the estimated PAT transforms are efficient for the normalisation of data coming from the same speakers for which the PAT transforms were trained and whose discrimination has to be maximised. Since there is not always enough data for each of the speakers to train reliable PAT transforms, it would be useful to develop adaptation techniques for the PAT transforms. Such techniques would allow the training of general PAT transforms on large databases with an elevated number of general speakers and following the adaptation to the specific recording

with a just a little amount of training data. Specifically, in the case of the semi-supervised on-line diarization, PAT transforms optimised for the involved speakers could be obtained in the off-line phase with the same amount of data meant for speaker models initialisation.

Appendix A

Diarisation du locuteur en temps réel pour les objets intelligents

Introduction

La diarisation du locuteur en temps réel vise à détecter "qui parle maintenant" dans un flux audio donné. La majorité des systèmes de diarisation en ligne proposés a mis l'accent sur des domaines moins difficiles, tels que la émission des nouvelles et discours en plénière, caractérisé par une faible spontanéité. La première contribution de cette thèse est le développement d'un système de diarisation du locuteur complètement un-supervisé et adaptatif en ligne pour les données de réunions qui sont plus difficiles et spontanées. En raison des hauts taux d'erreur de diarisation, une approche semi-supervisé pour la diarisation en ligne, ou les modèles des interlocuteurs sont initialisés avec une quantité modeste de données étiquetées manuellement et adaptées par une incrémentielle maximum a-posteriori adaptation (MAP) procédure, est proposée. Les erreurs obtenues peuvent être suffisamment bas pour supporter des applications pratiques.

La deuxième partie de la thèse aborde le problème de la normalisation phonétique pendant la modélisation des interlocuteurs avec petites quantités des données. Tout d'abord, Phone Adaptive Training (PAT), une technique récemment proposé, est évalué et optimisé au niveau de la modélisation des interlocuteurs et dans le cadre de la vérification automatique du locuteur (ASV) et est ensuite développée vers un système entièrement un-supervise en utilisant des transcriptions de classe acoustiques générées automatiquement, dont le nombre est contrôlé par analyse de l'arbre de régression. PAT

offre des améliorations significatives dans la performance d'un système ASV iVector, même lorsque des transcriptions phonétiques précises ne sont pas disponibles. Enfin, une première tentative de combinaison de PAT et diarisation semi-supervisé en ligne confirme le potentiel de PAT dans l'amélioration de la modélisation des interlocuteurs en temps réel et motive plus de recherche dans cette direction.

A.1 Diarisation en-ligne un-supervisé

Cette section implique le développement d'un système de diarisation un-supervisé et adaptatif en ligne pour les données de réunions. Contrairement à la plupart des ouvrages de littérature qui se concentrent sur les nouvelles de radiodiffusion et les scénarios de discours en plénière, le système proposé est plutôt développé et optimisé pour les données de réunions. De nos jours, les enregistrements de réunions représentent les données les plus difficiles disponibles pour développer un système de diarisation en ligne pour les applications en temps réel.

Le système développé est basé sur l'introduction séquentielle et l'adaptation des modèles de locuteurs au moyen d'un algorithme d'adaptation MAP séquentiel. La performance du système est évaluée par des expérimentes où différentes durées de segment de parole T_S et différentes tailles de modèle sont utilisées.

Bien que les performances du système corresponde aux performances d'autres systèmes de diarisation en ligne présentés dans la littérature qui traite des données moins difficiles, les taux d'erreur obtenus mettent en évidence le défi que pose le développement d'un système de diarisation en ligne efficace et apte à supporter des applications pratiques.

A.1.1 Implémentation du système

Le système de diarisation des interlocuteurs en ligne un-supervisé développé pour la diarisation des réunions est illustré dans la Figure A.1. Il est basé sur l'approche de base top-down de la diarisation hors ligne décrite au Chapitre 3, Section 3.4 et l'approche de diarisation en ligne rapportée dans [3]. Mis à part la modélisation d'arrière-plan, il existe trois étapes: (i) extraction d'observations acoustiques; (ii) la détection de l'activité de la parole et (iii) la classification en ligne.

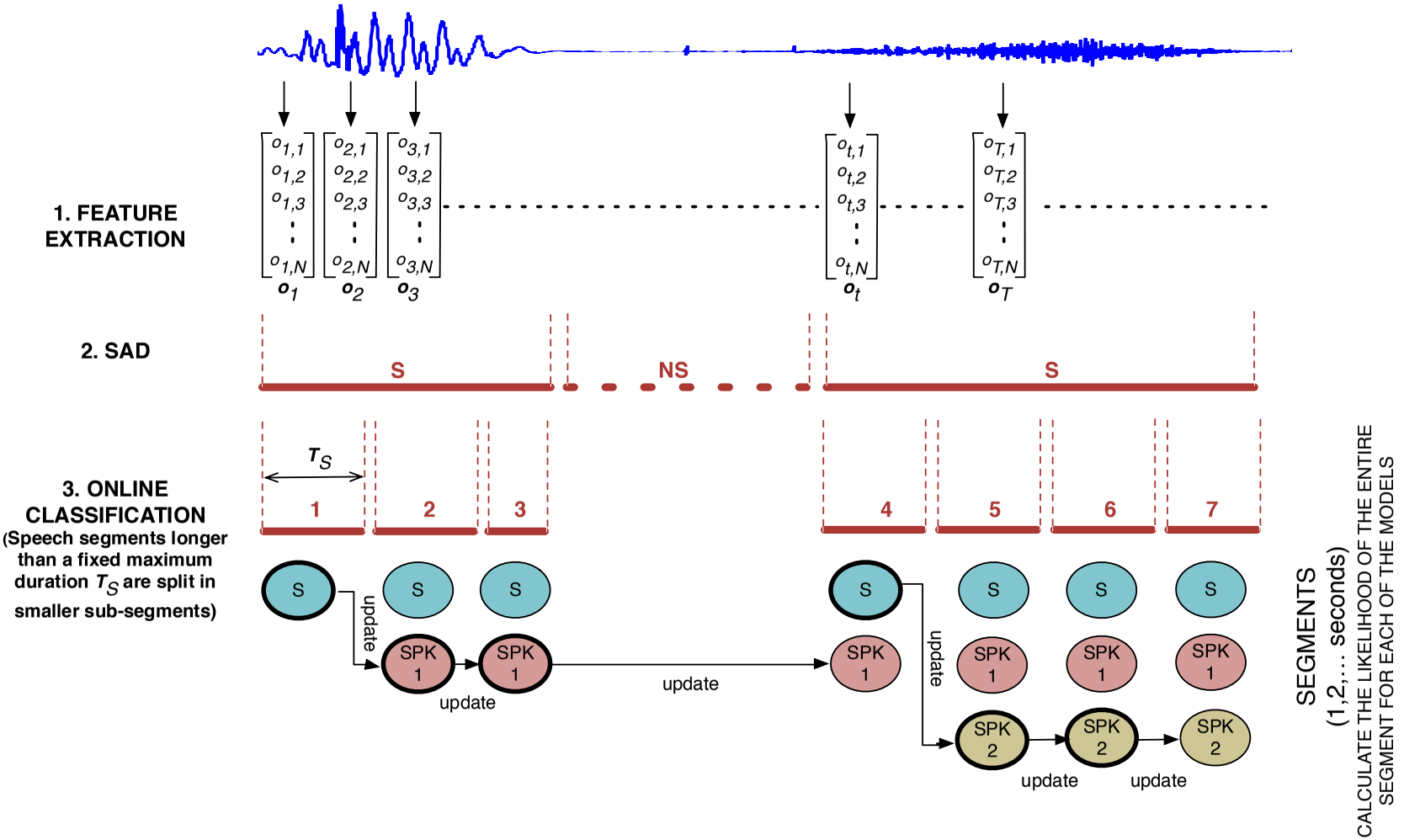


Fig. A.1 Une illustration du système de diarisation en ligne un-supervisé.

Détection de l'activité de la parole

Le flux audio est d'abord paramétré par une série d'observations acoustiques $\mathbf{o}_1, \ldots, \mathbf{o}_T$. Critiquement, pour tous les instants $\tau \in 1, \ldots, T$ seules les observations pour $t < \tau$ sont utilisées pour la diarisation. Les segments non vocaux sont supprimés en fonction de la sortie d'un détecteur d'activité de parole conventionnel basé sur le modèle dérivé du système de diarisation top-down de base décrit dans le Chapitre 3, Section 3.4. Les segments de parole restants sont ensuite divisés en sous-segments plus petits dont la durée ne dépasse pas une durée maximale fixée a priori T_S. Les valeurs plus élevées de T_S impliquent une latence plus élevé. La classification en ligne est ensuite appliquée en séquence à chaque segment.

Classification en ligne

Les segments de parole sont attribués à un modèle de interlocuteur existant ou un nouveau modèle de interlocuteur est créé. Cette procédure est contrôlée avec un universal background model (UBM) appelée s_0 qui est formé sur des données externes. Les nouveaux modèles de interlocuteurs sont introduits dans l'inventaire des interlocuteurs, si le segment actuel i génère une vraisemblance logarithmique plus élevée par rapport à l'UBM que par un ensemble de modèles de interlocuteur s_j, où $j = 1, \ldots, N$ Et où N indique le nombre de locuteurs dans l'hypothèse actuelle. Les segments sont attribués selon:

$$s_j = \arg\max_{l \in (0,\ldots,N)} \sum_{k=1}^{K} \mathcal{L}\left(\mathbf{o}_k | s_l\right) \tag{A.1}$$

où $\mathbf{o}_k$ est la k-th observation acoustique dans le segment i, K représente le nombre de observations acoustiques dans le i-th segment et où $\mathcal{L}\left(\mathbf{o}_k | s_l\right)$ désigne la vraisemblance logarithmique de la k-th observation acoustique dans le segment i donné le modèle GMM s_l. Si le segment est attribué à s_0, un nouveau modèle de interlocuteur s_{N+1} est obtenu par une adaptation MAP du modèle UBM s_0 en utilisant les observations acoustiques contenues dans le segment i. Le segment i est ensuite étiqueté selon le nouvel interlocuteur introduit et N est augmenté de un. Lorsqu'un segment est attribué à un interlocuteur existant, le modèle correspondant est adapté par une adaptation MAP séquentielle, illustré en Figure A.2. Le segment est ensuite marqué selon l'orateur reconnu j par Eq. A.1.

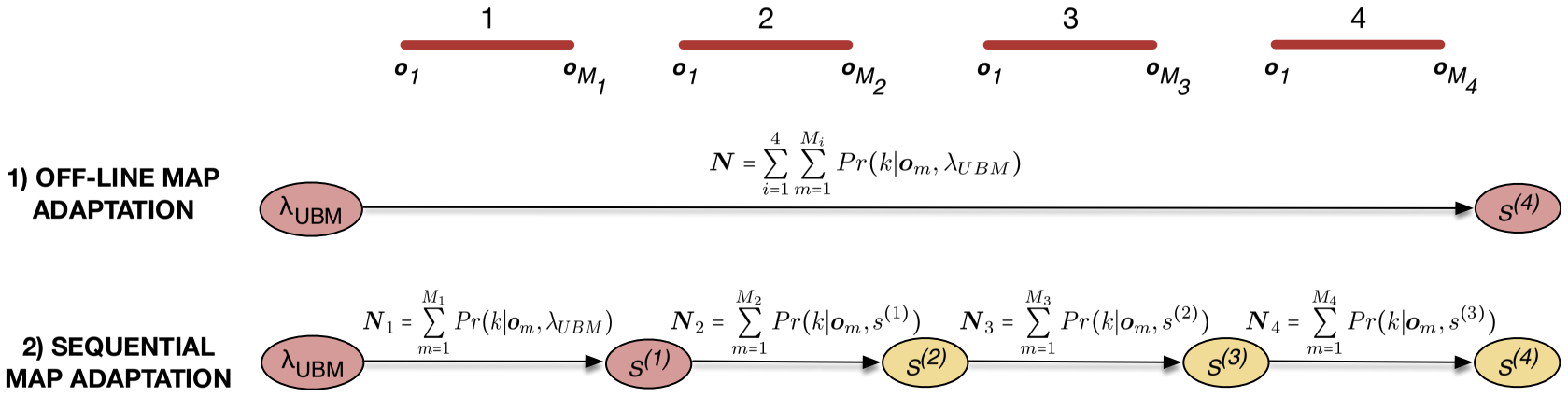


Fig. A.2 Une comparaison de l'adaptation MAP classique et l'adaptation MAP séquentielle avec quatre segments de parole d'un interlocuteur.

A.1.2 Evaluation de la performance

La performance du système de diarisation en ligne un-supervisé a été évaluée en analysant le DER global en fonction de la durée maximale du segment T_S et de la taille des modèles de interlocuteurs.

Des expérimentes ont été réalisées pour des durées de segments maximales de $0,25, 0,5, 1, \ldots, 10$ secondes et différentes tailles de modèles UBM: 8, 16, 32, 64 et 128 composants gaussiens.

Les parcelles à gauche dans les figures A.3, A.4 et A.5 illustrent les performances de diarisation en ligne en termes de DER globale en fonction de la durée du segment T_S et la taille du modèle pour les ensembles de données RTdev, RT07 et RT09, respectivement. La taille optimale du modèle est soit 32 ou 64 composants gaussiens, la plus grande taille du modèle étant la plus uniforme dans les trois ensembles de données. Dans tous les cas, il est possible de constater que, à mesure que la taille du modèle augmente, les performances se détériorent davantage et probablement en raison du manque de données suffisantes pour un apprentissage et une adaptation fiables des modèles de interlocuteurs. La durée maximale optimale du segment T_S pour tous les cas est d'environ 3 ou 4 secondes. Initialement, le DER tend à diminuer à mesure que la durée du segment augmente. À mesure que la taille du segment augmente au-delà de l'optimum, le DER global empire jusqu'à ce qu'il se stabilise. Ceci est probablement dû au fait que la plupart des segments de discours après le processus SAD sont déjà plus courts que la durée maximale du segment T_S. Dans les trois ensembles de données, le DER minimum est compris entre 40 % et 45 %. Il s'agit d'un taux d'erreur élevé, mais pas différent de celui rapporté dans les travaux antérieurs effectués sur de données de radiodiffusion, par exemple [3]. Les taux d'erreur de diarisation élevés peuvent être causés par l'initialisation des modèles de interlocuteurs grâce à l'adaptation MAP du modèle UBM avec des segments de parole trop courts. Les modèles de interlocuteurs initiaux ne sont pas suffisamment discriminants pour classer de manière fiable les segments de parole entrants. Bien que l'application de l'adaptation et la re-segmentation améliorent la performance, elles introduiraient également une latence et une complexité de calcul supplémentaires non conformes à la diarisation en ligne.

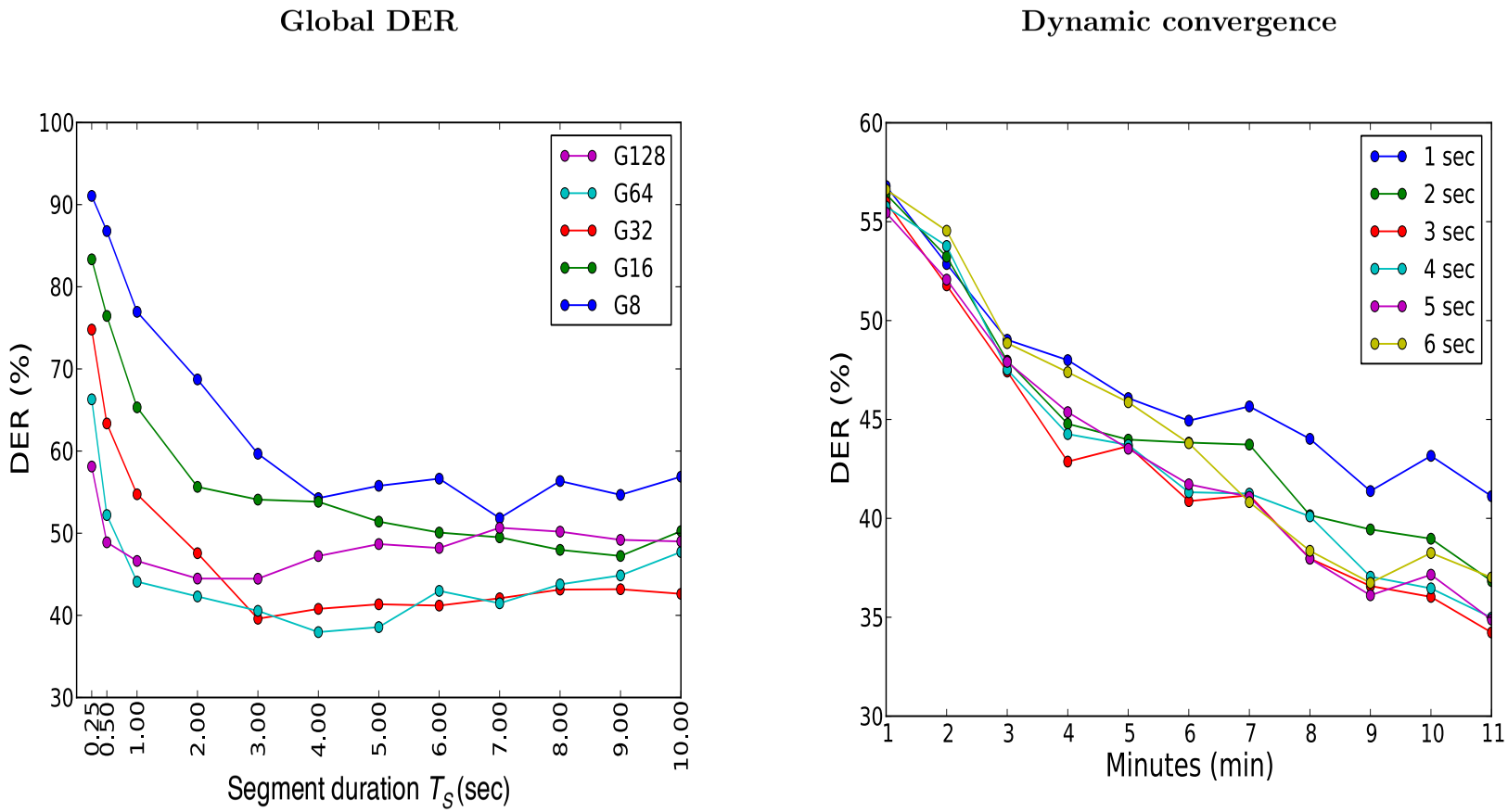


Fig. A.3 Les résultats sont affichés pour l'ensemble de données RTdev. **Diagrammes à gauche:** une illustration du DER global en fonction de la durée du segment T_S (0.25,0.5,1-10 sec) et pour différentes tailles de modèle (8-128).
Diagrammes à droite: une illustration de la convergence dynamique du DER en fonction du temps T_i.

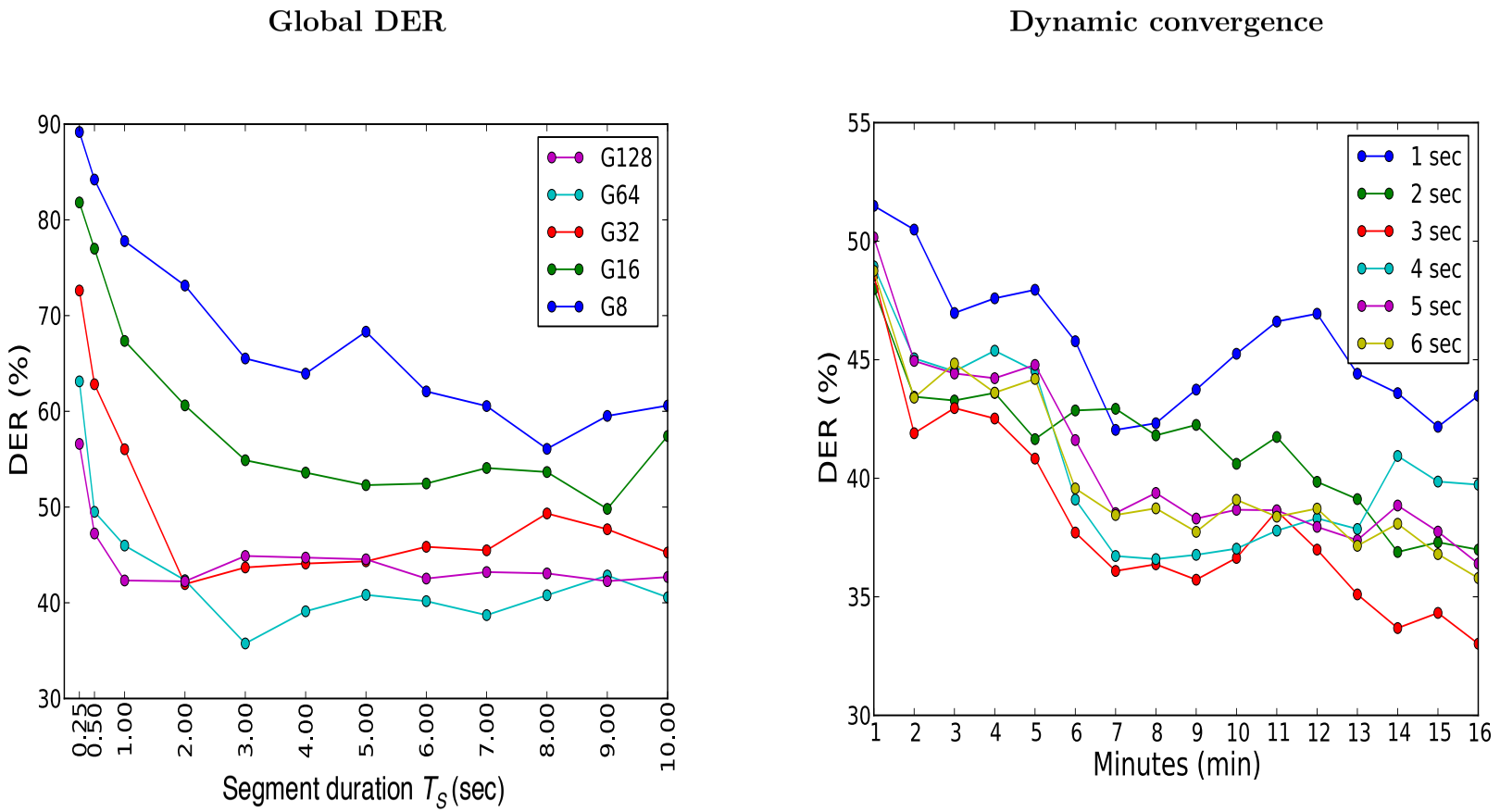


Fig. A.4 Les résultats sont affichés pour l'ensemble de données RT07. **Diagrammes à gauche:** une illustration du DER global en fonction de la durée du segment T_S (0.25,0.5,1-10 sec) et pour différentes tailles de modèle (8-128).
Diagrammes à droite: une illustration de la convergence dynamique du DER en fonction du temps T_i.

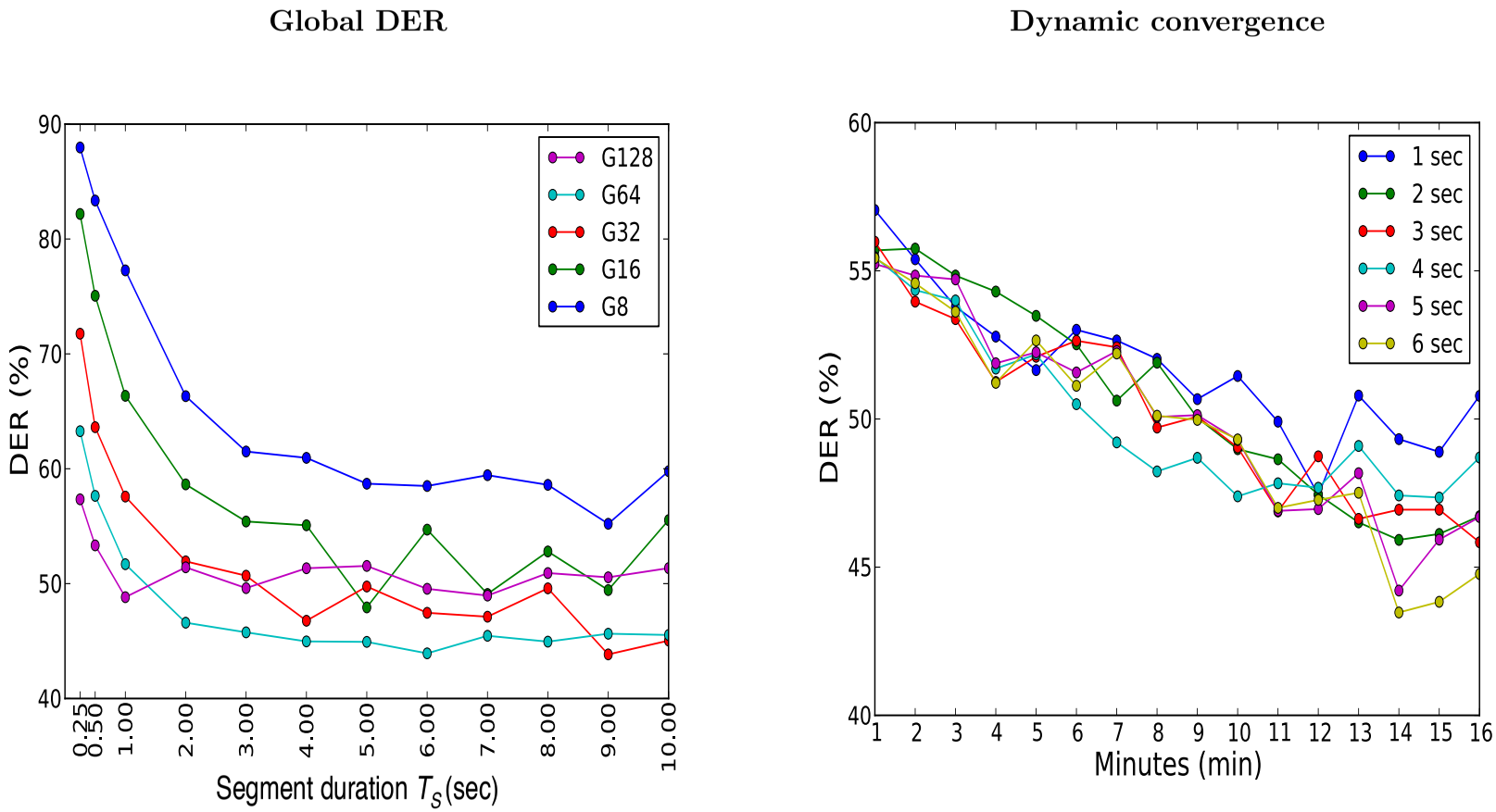


Fig. A.5 Les résultats sont affichés pour l'ensemble de données RT09. **Diagrammes à gauche:** une illustration du DER global en fonction de la durée du segment T_S (0.25,0.5,1-10 sec) et pour différentes tailles de modèle (8-128).
Diagrammes à droite: une illustration de la convergence dynamique du DER en fonction du temps T_i.

A.2 Semi-supervised on-line diarisation

Bien que des approches semi-supervisées aient été signalées précédemment pour la diarisation hors ligne [64], cette section concerne le développement d'un nouveau système de diarisation en ligne semi-supervisé. Le nouveau système exploite de courtes quantités de parole libellées pour l'initialisation supervisés des modèles de locuteurs. Le reste du processus reste entièrement un-supervisé. Le principal objectif des travaux présentés dans cette section est de déterminer la quantité de la parole libellées manuellement afin de fournir des performances satisfaisantes. La deuxième contribution de ce travail se rapporte plutôt à une maximum a-posteriori adaptation (MAP) procédure incrémentielle pour l'adaptation des modelés en ligne, qui s'avère déterminante dans la production de faibles taux d'erreur de diarisation. Cette procédure est illustrée en Figure A.6.

A.2.1 Implémentation du système

Le système de diarisation en ligne semi-supervisé proposé est illustré dans la Figure A.7. Il est basé sur l'approche top-down de la diarisation hors ligne reportée dans le Chapitre 3, Section 3.4 et l'approche de diarisation en ligne un-supervisée décrite en Chapitre 4.

Le système se caractérise par quatre étapes: (i) extraction des observations acoustiques; (ii) l'initialisation des modèles de interlocuteurs hors ligne; (iii) la détection de l'activité de la parole et (iv) la classification en ligne.

Extraction des observations acoustiques

Chaque flux audio est paramétré pour la première fois par une série d'observations acoustiques $\mathbf{o}_1, \ldots, \mathbf{o}_T$. Critiquement, pour n'importe quel instant $\tau \in 1, \ldots, T$ seules les observations pour $t < \tau$ sont utilisées pour la diarisation.

Initialisation des modèles de interlocuteurs hors ligne

Une brève phase de table ronde dans laquelle chaque orateur se présente est utilisée pour initialiser des modèles de interlocuteurs. Le premier T_{SPK} secondes de la parole active pour chaque orateur est mis de côté pour l'initialisation des modèles. Un inventaire $\tilde{\Delta}$ des modèles de interlocuteur s_j, où $j = 1, \ldots, M$, avec M indiquant le nombre de interlocuteurs lors d'une réunion, est ensuite formé en utilisant un

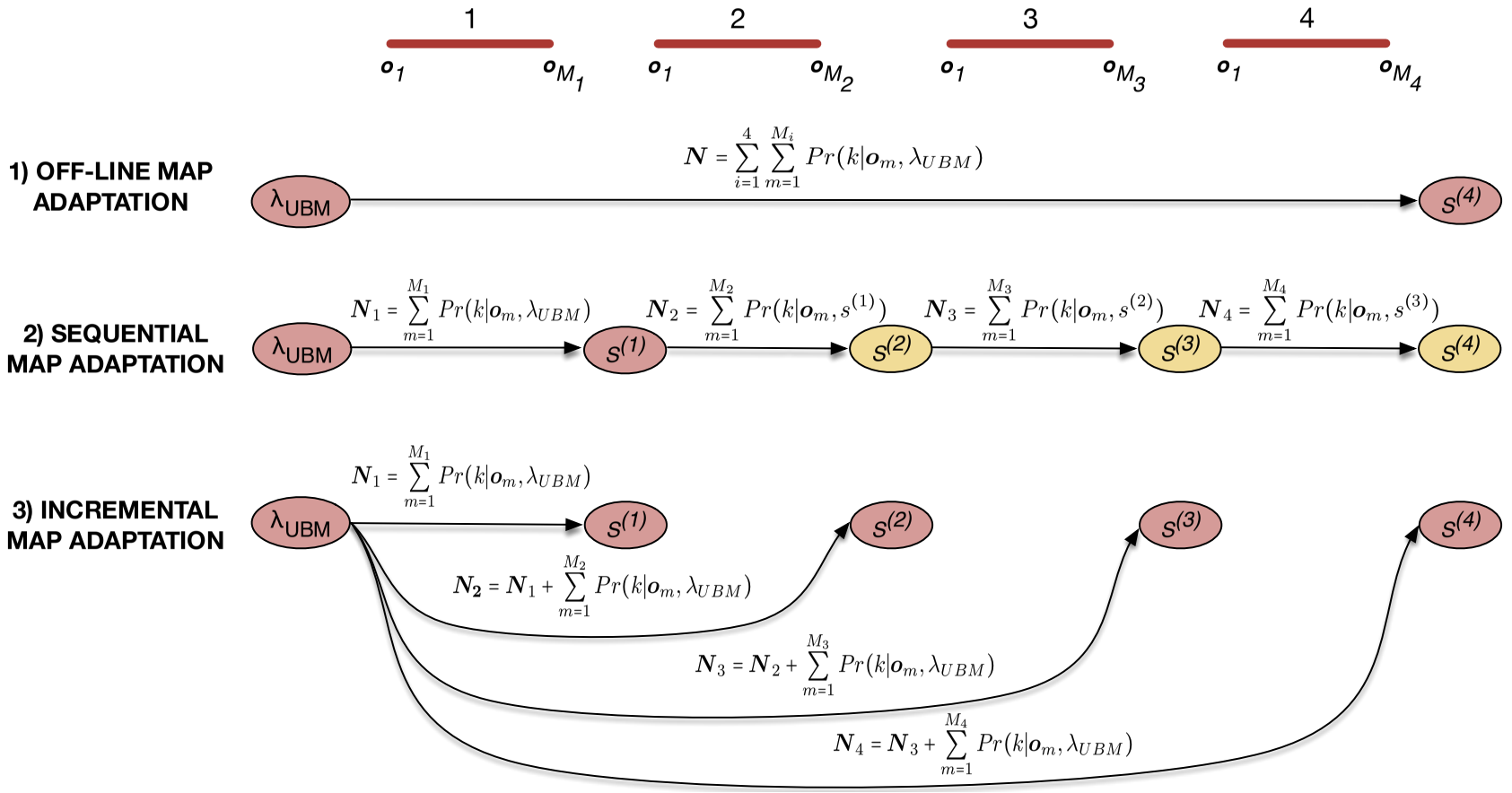


Fig. A.6 Une comparaison de l'adaptation MAP classique, l'adaptation MAP séquentielle et l'adaptation MAP incrémentielle pour quatre segments de parole.

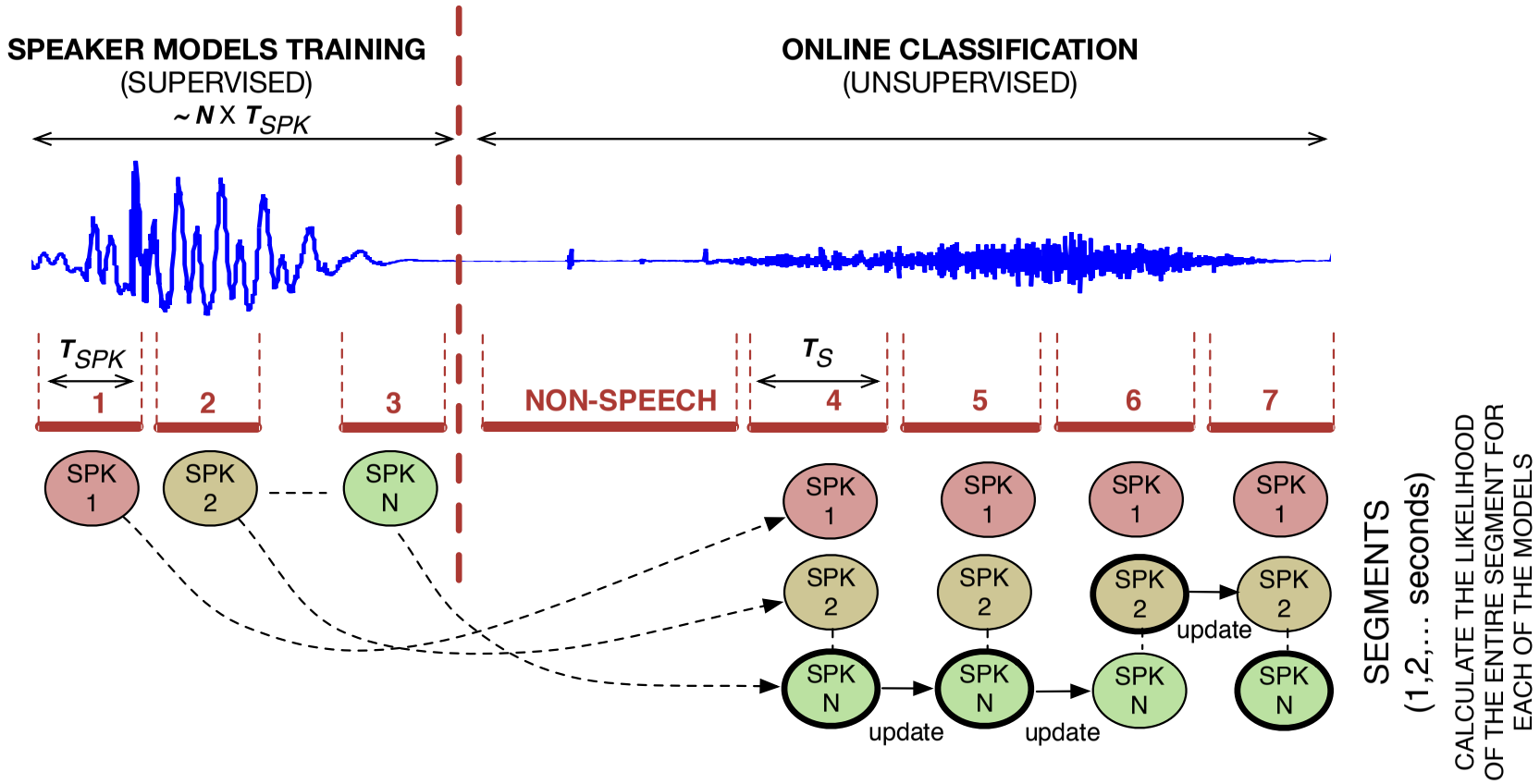


Fig. A.7 Une illustration du système de diarisation semi-supervisé en temps réel.

certain durée des données T_{SPK} pour chaque orateur. Les modèles de interlocuteurs sont obtenues par MAP adaptation de l'UBM en utilisant les données. Pour chaque modèle de interlocuteur s_j, les statistiques suffisantes $N_1^{(j)}, \boldsymbol{F}_1^{(j)}$ et $\boldsymbol{S}_1^{(j)}$ obtenu pendant l'adaptation MAP sont stockés afin d'être utilisés pendant la phase de classification en ligne pour mettre à jour les modèles de interlocuteur. L'ensemble résultant de modèles de interlocuteurs est ensuite utilisé pour diariser les segments de parole restants d'une manière un-supervisée.

Détection de l'activité de la parole et classification en ligne

Les segments non vocaux sont supprimés en fonction de la sortie d'un détecteur d'activité de la parole conventionnel (SAD) basé sur le modèle dérivé du système de diarisation top-down décrit dans le Chapitre 3, Section 3.4.

Les segments de parole restants sont ensuite divisés en sous-segments plus petits dont la durée ne dépasse pas une durée maximale fixée a priori T_S. Les valeurs plus élevées de T_S impliquent un système de latence plus élevé. La diarisation en ligne est ensuite appliquée en séquence à chaque sous-segment. La séquence des interlocuteurs optimisée $\tilde{S}$ et la segmentation $\tilde{G}$ sont obtenues en attribuant successivement chaque segment i à l'un des modèles de interlocuteur M selon:

$$s_j = \underset{l\in(1,\ldots,M)}{\arg\max} \sum_{k=1}^{K} \mathcal{L}\left(\mathbf{o}_k | s_l\right) \tag{A.2}$$

où $\mathbf{o}_k$ est la k-th observation acoustique du segment i, K représente le nombre des observations acoustiques dans le i-th segment et où $\mathcal{L}\left(\mathbf{o}_k | s_l\right)$ désigne la vraisemblance logarithmique de la k-th observation du segment i étant donné le modèle de interlocuteur s_l. Le segment est ensuite marqué d'après l'interlocuteur reconnu j par (A.2). Le modèle de interlocuteur mis à jour s_j est obtenu par une adaptation MAP séquentielle ou incrémentielle comme illustré dans la Figure A.6.

A.2.2 Evaluation de la performance

Afin d'évaluer la performance du système de diarisation semi-supervisé en ligne proposé, les DER globaux moyens sont évalués en fonction de la quantité de données de formation étiquetées T_{SPK} et de la durée maximale du segment T_S.

Tout d'abord, pendant la phase hors ligne et supervisée, les modèles de interlocuteur s_j sont formés à l'aide de quantités croissantes de données étiquetées T_{SPK} de durée $1, \ldots, 39$ secondes. Le modèle UBM général est composé par 64 composants gaussian.

La diarisation en ligne est effectuée en utilisant une durée de segment maximale différente de T_S avec $T_S = 0.25, 0.5, 1, 2, 3, 4$. Les valeurs supérieures de T_S impliquent une latence plus élevée du système. La performance du système est évaluée en utilisant la procédure d'adaptation MAP séquentielle et incrémentielle afin de prouver la meilleure efficacité de ce dernier en fournissant un taux d'erreur de diarisation inférieur.

Résultats dans les figures A.8, A.9 et A.10 illustrent la variation de DER par rapport à la quantité de données de formation des modèles des interlocuteurs T_{SPK} pour les ensembles de données, RTdev, RT07 et RT09, respectivement. Les parcelles à gauche illustrent les performances pour l'adaptation MAP séquentielle alors que les parcelles à droite correspondent à une adaptation MAP incrémentielle. Dans chaque parcelle, différents profils illustrent les performances pour une gamme de latences T_S.

La première observation des figures A.8, A.9 et A.10 indique que la performance du système semi-supervisé de diarisation en ligne peut dépasser celle de baseline système, le système de diarisation hors ligne (illustré par des lignes horizontales et pointillées). Dans le cas de l'adaptation MAP séquentielle, cela est réalisé pour l'ensemble de données RTdev, par exemple, lorsque les modèles de interlocuteurs sont formées avec $T_{SPK} = 9$ de secondes de données de formation lors de l'utilisation d'une taille de segment / latence de $T_S = 4$ secondes. Avec la même taille de segment, les performances de base pour les ensembles de données RT07 et RT09 sont dépassées en utilisant aussi peu que $T_{SPK} = 5$ et 3 secondes respectivement.

En général, des DER inférieurs sont obtenus avec une plus grande quantité de données de formation, par exemple, un DER de 12,5 % est obtenu avec $T_{SPK} = 9$ de secondes de données de formation pour l'ensemble de données RT07 et 15 % avec 17 secondes de données de formation pour l'ensemble de données RT09, toutes deux avec des latences de $T_S = 3$ secondes.

En tournant à côté des résultats pour la MAP incrémentielle illustrée dans les traits droites de Figures A.8, A.9 et A.10 il est immédiatement évident que les performances sont significativement meilleures que les performances avec la MAP séquentielle. Ici, la performance de diarisation baseline, hors ligne est dépassée avec aussi peu que $T_{SPK} = 5$ de secondes de données de formation pour l'ensemble de données RTdev et

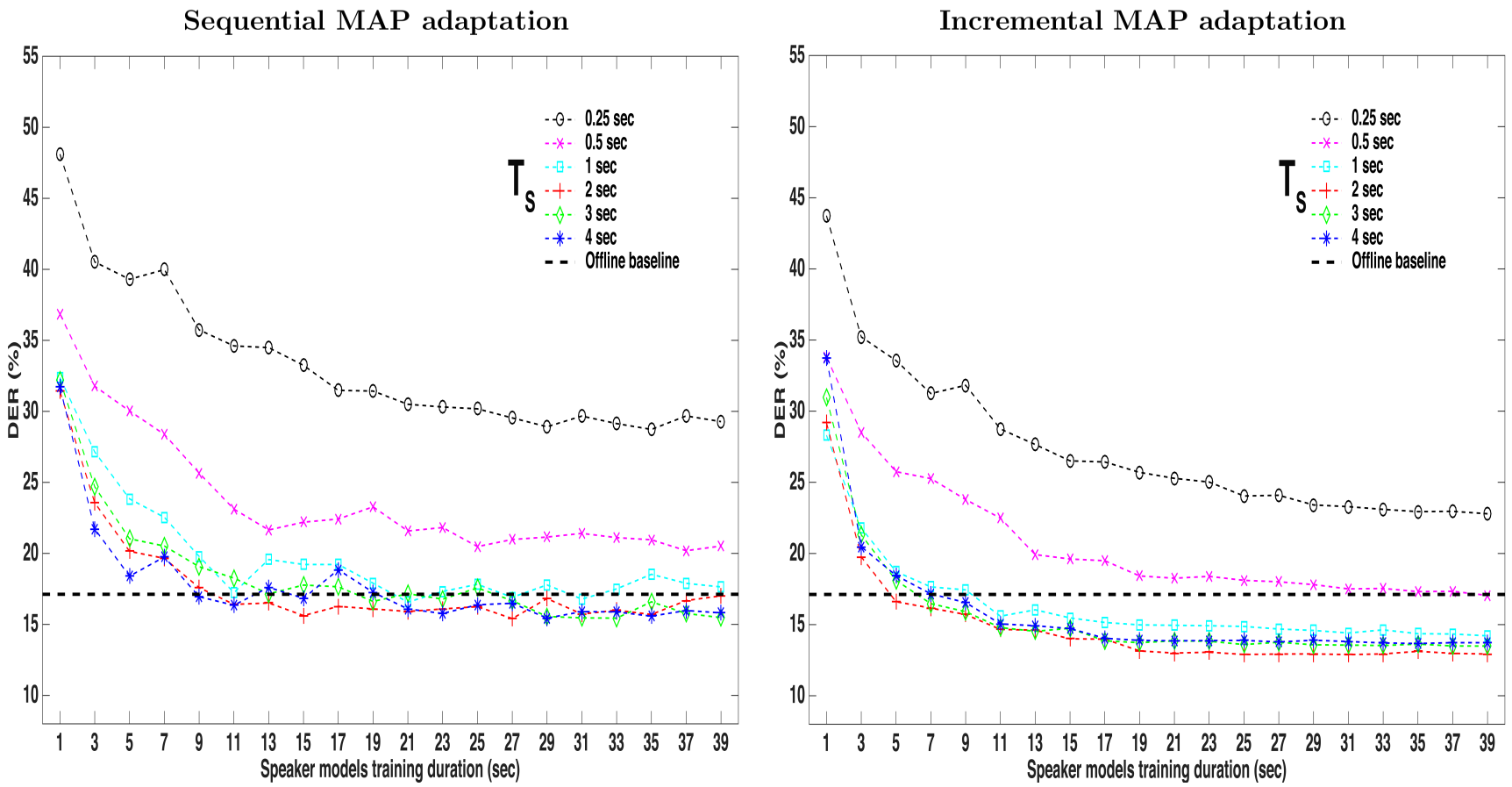


Fig. A.8 Une illustration de DER pour le système de diarisation en ligne semi-supervisé en fonction de la durée de formation des modèles des interlocuteurs T_{SPK} et pour différentes durées / latences maximales de segments T_S. Résultats affichés pour l'ensemble de données de développement RTdev en utilisant l'adaptation séquentielle MAP (à gauche) et l'adaptation MAP incrémentielle (à droite). La ligne horizontale et pointillée indique la performance du système de diarisation hors ligne.

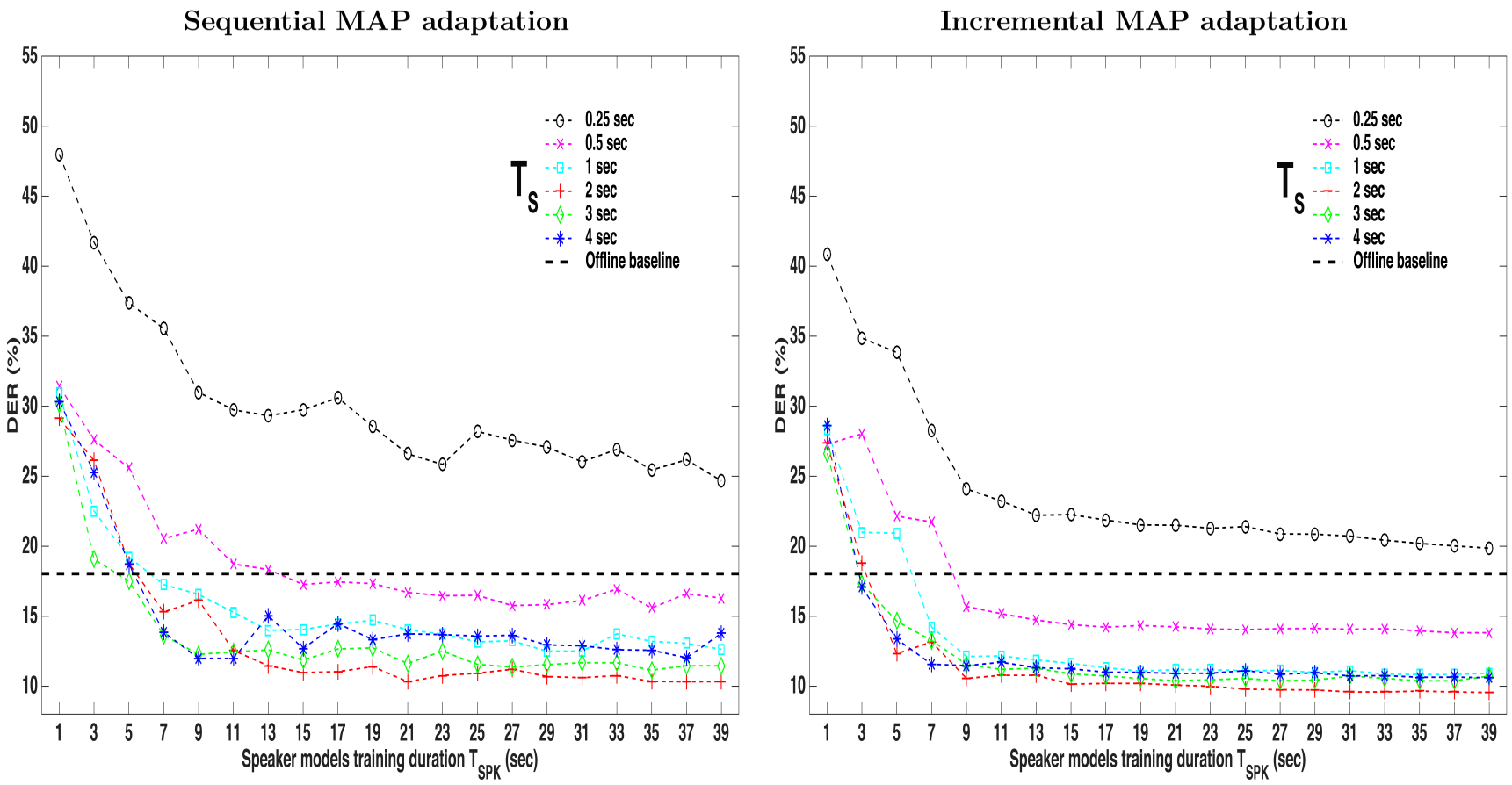


Fig. A.9 Une illustration de DER pour le système de diarisation en ligne semi-supervisé en fonction de la durée de formation des modèles des interlocuteurs T_{SPK} et pour différentes durées / latences maximales de segments T_S. Résultats affichés pour l'ensemble de données d'évaluation RT07 en utilisant l'adaptation séquentielle MAP (à gauche) et l'adaptation MAP incrémentielle (à droite). La ligne horizontale et pointillée indique la performance du système de diarisation hors ligne.

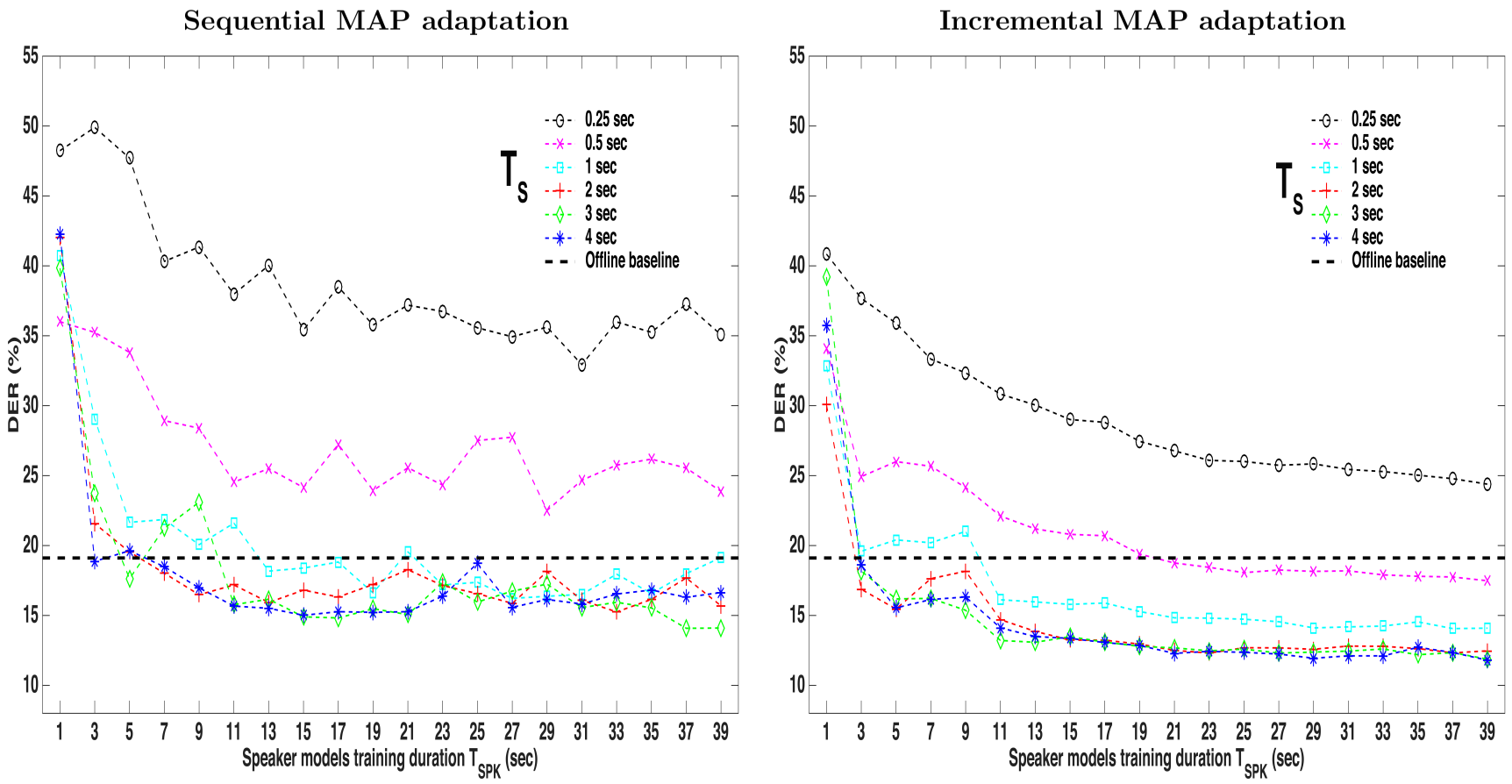


Fig. A.10 Une illustration de DER pour le système de diarisation en ligne semi-supervisé en fonction de la durée de formation des modèles des interlocuteurs T_{SPK} et pour différentes durées / latences maximales de segments T_S. Résultats affichés pour l'ensemble de données d'évaluation RT09 en utilisant l'adaptation séquentielle MAP (à gauche) et l'adaptation MAP incrémentielle (à droite). La ligne horizontale et pointillée indique la performance du système de diarisation hors ligne.

$T_{SPK} = 3$ secondes dans le cas de RT07 et RT09, tous avec une latence aussi faible que $T_S = 2$ secondes. Encore une fois, des DER inférieurs sont obtenus avec une plus grande quantité de données de formation, aussi bas que 10% pour l'ensemble de données RT07 et 12,5% pour l'ensemble de données RT09. En général, des DER inférieurs sont obtenus avec une plus grande quantité de données de formation, par exemple, un DER de 12,5% est obtenu avec $T_{SPK} = 9$ de secondes de données de formation pour l'ensemble de données RT07 et 15% avec 17 secondes de données de formation pour l'ensemble de données RT09, toutes deux avec des latences de $T_S = 3$ secondes.

A.3 Phone adaptive training

Phone adaptive training (PAT) est un algorithme récemment introduit [4] dont le but est de normaliser la variation du phonèmes dans la diarisation des interlocuteurs en projetant les observations acoustiques dans un nouvel espace dans lequel la discrimination de phonèmes est minimisée alors que la discrimination des interlocuteurs est maximisée. Alors que PAT fonctionne au niveau des observations acoustiques et cible la modélisation améliorée des interlocuteurs, son utilisation dans un cadre de diarisation des interlocuteurs permet une optimisation quelque peu gênante.

La première contribution sur PAT est l'évaluation et l'optimisation de PAT indépendamment des complexités convolutives de la diarisation des locuteurs et dans des conditions strictement contrôlées. Au moyen des expérimentes oracle de vérification automatique du locuteur (ASV), la performance de PAT est analysée lorsqu'elle est appliquée à un système ASV texte indépendant de courte durée en fonction de la complexité du modèle et pour des quantités variables de données de formation, en utilisant l'ensemble de données TIMIT, qui est étiqueté manuellement au niveau du téléphone.

La deuxième contribution consiste en nos efforts pour développer PAT dans un système totalement un-supervisé. Les contributions comprennent une approche de la transcription automatique des classes acoustiques au moyen de l'analyse de l'arbre de régression. Comme pour le premier travail, la performance de PAT est analysée en fonction de la complexité du modèle et pour la variation des quantités de données de formation. Les expérimentes montrent que PAT fonctionne bien, même lorsque le nombre de classes acoustiques est réduit bien en dessous du nombre des phonèmes, ce qui réduit le besoin de transcriptions phonétiques précises.

A.3.1 Oracle ASV expérimentes

Comme illustré dans la Figure A.11, PAT est appliqué aux observations acoustiques originales $\boldsymbol{O}_{s,p}$ selon les transcriptions phonétiques originales. Les observations acoustiques transformées $\tilde{\boldsymbol{O}}_{s,p}$ sont ensuite utilisées pour les expérimentes ASV tandis que les observations acoustiques originales $\boldsymbol{O}_{s,p}$ pour les expérimentes ASV de base. La différence entre les taux d'erreur égaux obtenus (EER) à partir des expérimentes ASV de base et des expérimentes ASV avec PAT est considérée comme une mesure de la

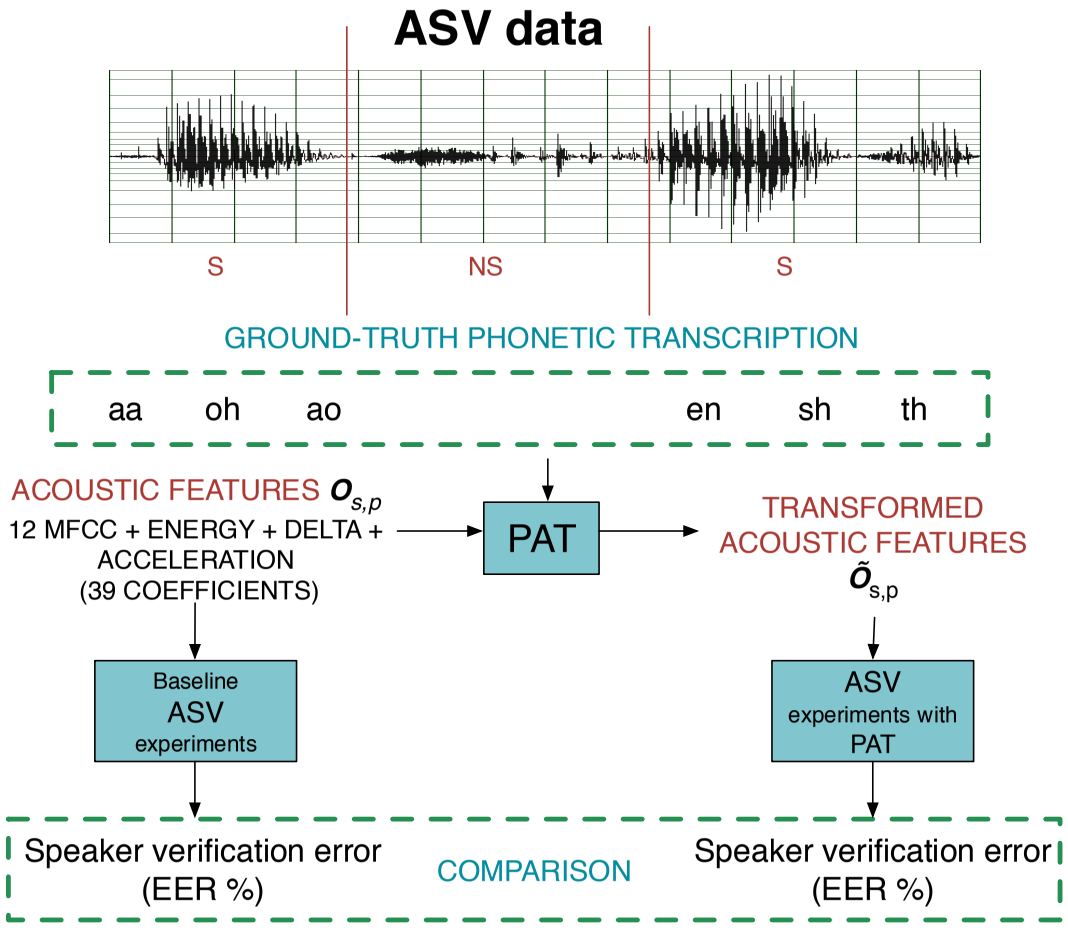


Fig. A.11 Une illustration de la configuration expérimentale de l'oracle ASV système.

(a) GMM-UBM

Number of sentences for speaker model training	Baseline (EER %)	Baseline + PAT (EER %)
1	4.2	3.6
3	1.8	1.0
5	0.6	0.6
7	0.6	0.6

(b) iVector-PLDA

Number of sentences for speaker model training	Baseline (EER %)	Baseline + PAT (EER %)
1	2.4	1.2
3	1.1	0.4
5	1.1	0.4
7	0.6	0.3

Table A.1 Une illustration des EER pour le GMM-UBM et les systèmes iVector-PLDA avec des quantités variables de données de formation. Les résultats sont affichés pour des tailles de modèles optimales dans chaque cas.

puissance discriminatoire des interlocuteurs dans le cas où les modèles de interlocuteurs sont formées avec des énoncés de courte durée et dans conditions optimales.

La table A.1 illustre un résumé des performances de PAT pour les systèmes ASV GMM-UBM et iVector-PLDA pour différentes quantités de données de formation. Les résultats correspondent à des tailles de modèles optimales dans chaque cas. Lorsque les modèles des interlocuteurs sont formés sur une seule phrase, le système iVector-PLDA de bas de base surpasse le système GMM-UBM de référence de 43 % relatif (EER de 4.2 % c.f. 2.4 %). Lorsque 7 phrases sont utilisées, les deux systèmes atteignent le même EER de base de 0,6 %. PAT mène à des performances meilleures ou équivalentes dans tous les cas. Lorsque les modèles des interlocuteurs sont appris avec une seule phrase, les EER de base diminuent à 3,6% et 1,2 % pour les systèmes GMM-UBM et iVector-PLDA respectivement. À noter, les plus grandes améliorations de la performance ASV

sont obtenues pour le système iVector-PLDA où la performance est améliorée de 50% par rapport à la quantité de données de formation.

A.3.2 Vers PAT un-supervisé

Dans des scénarios réels, des transcriptions phonétiques fiables sont rarement disponibles et difficiles à obtenir. Dans cette section, nous présentons nos efforts pour développer PAT dans un système totalement unsupervisé. La transcription automatique des classes acoustiques s'effectue au moyen d'un système de reconnaissance des classes acoustiques dont la sortie est utilisée pour estimer les transformées PAT. Comme illustré dans la Figure A.12, les données destinées à l'estimation de l'UBM à partir de l'ensemble de données TIMITubm sont utilisées pour déterminer de 5 à 38 classes acoustiques au moyen d'une analyse d'arbre de régression binaire. Pour chacune des classes acoustiques, un modèle HMM est formé. Ces modèles sont ensuite introduits dans un reconnaissance automatique des classes acoustiques afin d'obtenir des transcriptions des classes acoustiques. PAT est ensuite appliqué sur les observations acoustiques originales des données de parole en fonction des transcriptions des classes acoustiques obtenues plutôt que des transcriptions phonétiques originales, comme indiqué dans la Section 6.3. Les observations transformées $\tilde{\boldsymbol{O}}_{s,p}$ sont ensuite utilisées pour les expérimentes ASV tandis que les observations acoustiques originales $\boldsymbol{O}_{s,p}$ pour les expérimentes ASV de base. La différence entre les taux d'erreur égaux obtenus (EER) des expérimentes ASV de base et des expérimentes ASV avec PAT est alors considérée comme une mesure pour quantifier la puissance discriminative de PAT dans le cas où les modèles de interlocuteurs sont inscrits avec des énoncés de courte durée et lorsque des transcriptions phonétiques générées automatiquement sont utilisées.

La figure A.13 illustre les performances de PAT pour le système GMM-UBM (barres claires) et les systèmes iVector-PLDA (barres ombrées) avec différents nombres de classes acoustiques. La complexité des deux systèmes est fixée à 64 composants. Bien que les enveloppes de profil ne soient pas convexes, probablement en raison du manque de données de formation, l'application de PAT donne de meilleures performances que les systèmes de base respectives (lignes horizontales solides et pointillées). Ces observations indiquent que PAT est bénéfique même sans transcriptions phonétiques fiables. Avec 15 et 25 classes acoustiques respectivement, l'amélioration relative des performances est de 18 pour le système GMM-UBM et 33% pour le système iVector-PLDA.

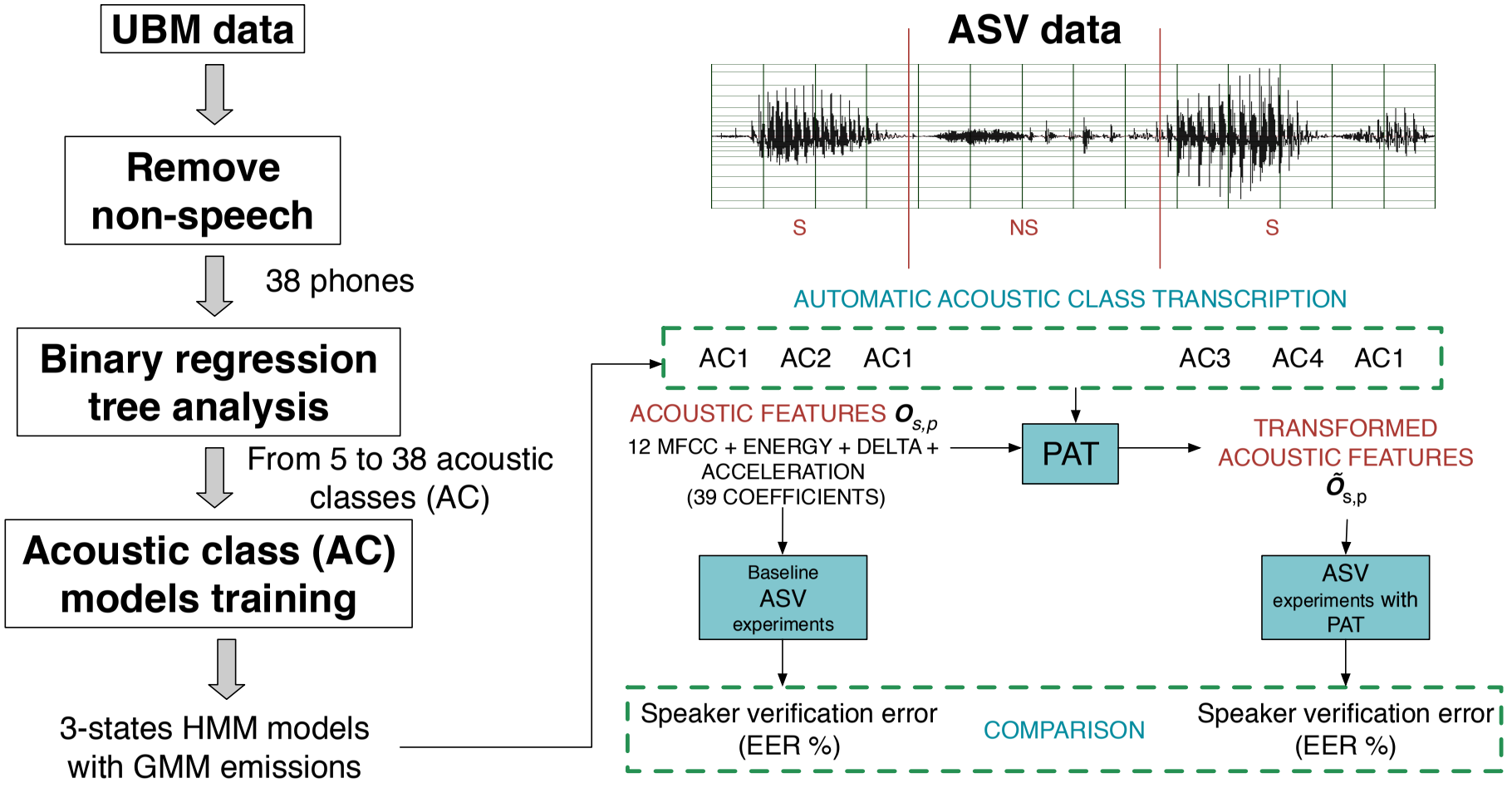


Fig. A.12 Une illustration de la configuration expérimentale pour PAT un-supervisé.

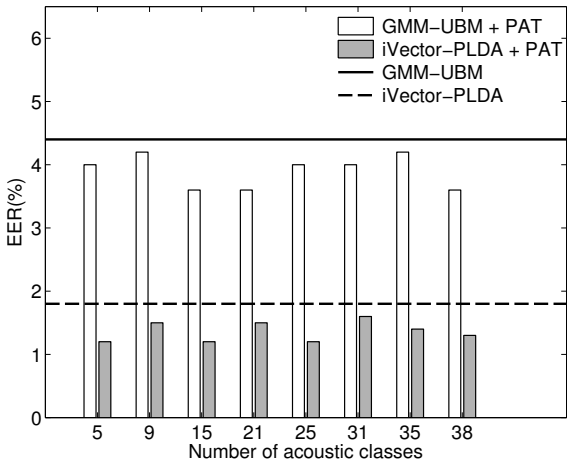


Fig. A.13 Une illustration de la performance ASV pour les systèmes GMM-UBM et iVector-PLDA avec 5 itérations de PAT pour différents nombres de classes acoustiques. Tous les modèles sont formés avec 1 phrase TIMIT et ils sont composés par 64 composants gaussian. Les performances de référence pour les systèmes GMM-UBM et iVector-PLDA sont représentées respectivement par les lignes horizontales solides et discontinues.

References

[1] K. Markov and S. Nakamura, "Never-ending learning system for on-line speaker diarization," in *IEEE Workshop Automatic Speech Recognition Understanding (ASRU)*, Dec 2007, pp. 699–704.

[2] K. Markov and S. Nakamura, "Improved novelty detection for online GMM based speaker diarization," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2008, pp. 363–366.

[3] J. T. Geiger, F. Wallhoff, and G. Rigoll, "GMM-UBM based open-set online speaker diarization.," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2010, pp. 2330–2333.

[4] S. Bozonnet, R. Vipperla, and N. Evans, "Phone adaptive training for speaker diarization," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2012.

[5] S.E. Tranter and D.A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech, Language Processing*, vol. 14, no. 5, pp. 1557–1565, Sept 2006.

[6] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. Audio, Speech, Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[7] M. Yamaguchi, M. Yamashita, and S. Matsunaga, "Spectral cross-correlation features for audio indexing of broadcast news and meetings," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2005.

[8] L. Shriberg, E.and Ferrer, A. Kajarekar, S.and Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, 2005.

[9] G. Friedland, O. Vinyals, Yan Huang, and C. Muller, "Prosodic and other long-term features for speaker diarization," *IEEE Trans. Audio, Speech, Language Processing*, vol. 17, no. 5, pp. 985–993, 2009.

[10] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," *Odyssey - The Speaker and Language Recognition Workshop*, 2001.

[11] S. Sinha, S. E. Tranter, M.J.F. Gales, and P.C. Woodland, "The Cambridge University March 2005 Speaker Diarisation System," *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2005.

[12] X. Zhu, C. Barras, L. Lamel, and J.-L. Gauvain, "Speaker diarization: From broadcast news to lectures," in *Machine Learning for Multimodal Interaction*, vol. 4299 of *Lecture Notes in Computer Science*, pp. 396–406. Springer Berlin Heidelberg, 2006.

[13] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059–1070, Dec 2010.

[14] J.M. Pardo, X. Anguera, and C. Wooters, "Speaker diarization for multiple-distant-microphone meetings using several sources of information," *IEEE Trans. Computer*, vol. 56, no. 9, pp. 1212–1224, Sept 2007.

[15] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Trans. Audio, Speech, Language Processing*, vol. 10, no. 7, pp. 504–516, Oct 2002.

[16] L. Lamel, L. Rabiner, A. Rosenberg, and J. Wilpon, "An improved endpoint detector for isolated word recognition," *IEEE Trans. Audio, Speech, Language Processing*, vol. 29, no. 4, pp. 777–785, Aug 1981.

[17] J. C. Junqua, B. Mak, and B. Reaves, "A robust algorithm for word boundary detection in the presence of noise," *IEEE Trans. Audio, Speech, Language Processing*, vol. 2, no. 3, pp. 406–412, Jul 1994.

[18] D. Istrate, C. Fredouille, S. Meignier, L. Besacier, and J.-F. Bonastre, "RT05 evaluation: Pre-processing techniques and speaker diarization on multiple microphone meetings," in *NIST 2005 Spring Rich Transcrition Evaluation Workshop*, 2005.

[19] D. A. van Leeuwen and M. Huijbregts, *Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006, Bethesda, MD, USA, May 1-4, 2006, Revised Selected Papers*, chapter The AMI Speaker Diarization System for NIST RT06s Meeting Data, pp. 371–384, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[20] X. Anguera, C. Wooters, B. Peskin, and M. Aguiló, *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005, Edinburgh, UK, July 11-13, 2005, Revised Selected Papers*, chapter Robust Speaker Segmentation for Meetings: The ICSI-SRI Spring 2005 Diarization System, pp. 402–414, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[21] C. Fredouille, S. Bozonnet, and N. Evans, "The LIA-EURECOM RT'09 speaker diarization system," in *RT09 NIST Rich Transcription Workshop*, Melbourne, Florida, USA, May 2009.

[22] E. Rentzeperis, A. Stergiou, C. Boukis, A. Pnevmatikakis, and L. C. Polymenakos, "The 2006 Athens information technology speech activity detection and speaker diarization systems," in *Int. Conf. Machine Learning for Multimodal Interaction*, Berlin, Heidelberg, 2006, MLMI'06, pp. 385–395, Springer-Verlag.

[23] A. Temko, D. Macho, and C. Nadeu, "Enhanced svm training for robust speech activity detection," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, April 2007, vol. 4, pp. IV–1025–IV–1028.

[24] J. Dines, J. Vepa, and T. Hain, "The segmentation of multi-channel meeting recordings for automatic speech recognition," in *Interspeech-ICSLP*, 2006.

[25] X. Anguera, M. Aguilo, C. Wooters, C. Nadeu, and J. Hernando, "Hybrid speech/non-speech detector applied to speaker diarization of meetings," in *Odyssey - The Speaker and Language Recognition Workshop*, June 2006.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[27] A. Reynolds, D., F. Quatieri, T., and B. Dunn, R., "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[28] N. Dehak, *Discriminative and generative approaches for long- and short-term speaker characteristics modeling: application to speaker verification*, Ph.D. thesis, Centre de recherche informatique de Montreal (CRIM), 2009.

[29] J.-F. Bonastre, P. M. Bousquet, D. Matrouf, and X. Anguera, "Discriminant binary data representation for speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2011, pp. 5284–5287.

[30] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system," in *Multimodal Technologies for Perception of Humans*, pp. 509–519. 2008.

[31] T. H. Nguyen, H. Sun, S. K. Zhao, S. Z. K. Khine, H. D. Tran, T. L. N. Ma, B. Ma, E. S. Chng, and H. Li, "The IIR-NTU speaker diarization systems for RT 2009," in *RT09 NIST Rich Transcription Workshop*, Melbourne, Florida, USA, May 2009.

[32] J. Ajmera, I. McCowan, and H. Bourlard, "Robust speaker change detection," *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 649–651, Aug 2004.

[33] D. Vijayasenan, F. Valente, and H. Bourlard, "An information theoretic approach to speaker diarization of meeting data," *IEEE Trans. Audio, Speech, Language Processing*, vol. 17, no. 7, pp. 1382–1393, Sept 2009.

[34] S. Shum, S. Dehak, E. Chuangsuwanich, D.A. Reynolds, and J.R. Glass, "Exploiting intra-conversation variability for speaker diarization," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2011, pp. 945–948.

[35] S. Shum, N. Dehak, and J. Glass, "On the use of spectral and iterative methods for speaker diarization," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2012, pp. 482–485.

[36] U. von Luxburg, "A tutorial on spectral clustering," 2007.

[37] S.H. Shum, N. Dehak, R. Dehak, and J.R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Trans. Audio, Speech, Language Processing*, vol. 21, no. 10, pp. 2015–2028, Oct 2013.

[38] F. Valente, *Variational bayesian methods for audio indexing*, Ph.D. thesis, EURECOM, September 2005.

[39] C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, 2006.

[40] J. Prazak and J. Silovsky, "Speaker diarization using plda-based speaker clustering," in *IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, Sept 2011, vol. 1, pp. 347–350.

[41] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE Int. Conf. Computer Vision (ICCV)*, 2007, pp. 1–8.

[42] D. Garcia-Romero and C. Y. Epsy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2011, pp. 249–252.

[43] G. Sell and D Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *IEEE Spoken Language Technology Workshop (SLT)*, Dec 2014, pp. 413–417.

[44] X. Anguera and J.-F. Bonastre, "Fast speaker diarization based on binary keys," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2011, pp. 4428–4431.

[45] H. Delgado, X. Anguera, C. Fredouille, and J. Serrano, "Improved binary key speaker diarization system," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Aug 2015, pp. 2087–2091.

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

[47] S. Meignier, J.-F. Bonastre, and S. Igounet, "E-HMM approach for learning and adapting sound models for speaker indexing," in *Odyssey - The Speaker and Language Recognition Workshop*, 2001.

[48] C. Fredouille and N. Evans, "The LIA RT'07 Speaker Diarization System," in *Int. Eval. Workshops CLEAR 2007 and RT 2007*, May 2007.

[49] S. Bozonnet, N.W.D. Evans, and C. Fredouille, "The LIA-Eurecom RT'09 speaker diarization system: enhancements in speaker modelling and cluster purification," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2010, pp. 4958–4961.

[50] M. Rouvier and S. Meignier, "A global optimization framework for speaker diarization," *Odyssey - The Speaker and Language Recognition Workshop*, 2012.

[51] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, Aug. 2013.

[52] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, Dec. 2006.

[53] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "A sticky hdp-hmm with application to speaker diarization," *Ann. Appl. Stat.*, vol. 5, no. 2A, pp. 1020–1056, 06 2011.

[54] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T. Nakatani, A. Nakamura, and J. Yamato, "Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera," *IEEE Trans. Audio, Speech, Language Processing*, vol. 20, no. 2, pp. 499–513, Feb 2012.

[55] Y. Aronowitz, H. Solewicz and O. Toledo-Ronen, "Online two speaker diarization," *Odyssey - The Speaker and Language Recognition Workshop*, 2012.

[56] W. Zhu and J. Pelecanos, "Online speaker diarization using adapted i-vector transforms," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, March 2016, pp. 5045–5049.

[57] T. Oku, S. Sato, A. Kobayashi, S. Homma, and T. Imai, "Low-latency speaker diarization based on bayesian information criterion with multiple phoneme classes," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, March 2012, pp. 4189–4192.

[58] C. Vaquero, O. Vinyals, and G. Friedland, "A hybrid approach to online speaker diarization," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2010, pp. 2638–2641.

[59] G. Friedland, A. Janin, D. Imseng, X. Anguera, L. Gottlieb, M. Huijbregts, M.T. Knox, and O. Vinyals, "The ICSI RT'09 speaker diarization system," *IEEE Trans. Audio, Speech, Language Processing*, vol. 20, no. 2, pp. 371–381, 2012.

[60] S. Bozonnet, *New insights into hierarchical clustering and linguistic normalization for speaker diarization*, Ph.D. thesis, EURECOM, 2012.

[61] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas, "Qualcomm-ICSI-OGI Features For ASR," in *Proc. ICSLP*, 2002, pp. 4–7.

[62] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium, Philadelphia*, 1993.

[63] S. Fernandez, A. Graves, and J. Schmidhuber, "Phoneme recognition in TIMIT with BLSTM-CTC," *CoRR*, 2008.

[64] D. Moraru, L. Besacier, and E. Castelli, "Using a priori information for speaker diarization," in *Odyssey - The Speaker and Language Recognition Workshop*, 2004.

[65] B. Fauve, N. Evans, N. Pearson, J.-F. Bonastre, and J. Mason, "Influence of task duration in text-independent speaker verification," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*. 2007, pp. 794–797, ISCA.

[66] R. J. Vogt, B. J. Baker, and S. Sridharan, "Factor analysis subspace estimation for speaker verification with short utterances," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2008, pp. 853–856.

[67] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "i-vector based speaker recognition on short utterances," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2011, pp. 2341–2344.

[68] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker adaptive training," in *Proc. Int. Conf. Spoken Language Processing (ICSLP)*, 1996, vol. 2, pp. 1137–1140.

[69] W. Shen and D. A. Reynolds, "Improving phonotactic language recognition with acoustic adaptation," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2007, pp. 358–361.

[70] L. Cheung-Chi, M. Bin, and L. Haizhou, "Parallel acoustic model adaptation for improving phonotactic language recognition," in *Odyssey - The Speaker and Language Recognition Workshop*, 2010, p. 41.

[71] S. Bozonnet, Dong Wang, N. Evans, and R. Troncy, "Linguistic influences on bottom-up and top-down clustering for speaker diarization," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2011, pp. 4424–4427.

[72] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2013, pp. 7649–7653.

[73] T. Hasan, R. Saeidi, J. H. L. Hansen, and D.A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2013, pp. 7663–7667.

[74] B. Fauve, N. Evans, and J. Mason, "Improving the performance of text-independent short duration SVM- and GMM-based speaker verification," in *Odyssey - The Speaker and Language Recognition Workshop*, 2008.

[75] A. K. Sarkar, D. Matrouf, P.-M. Bousquet, and J.-F. Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification.," in *Proc. Ann. Conf. International Speech Communication Association (INTERSPEECH)*, 2012.

[76] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[77] Haoze Lu, H. Okamoto, M. Nishida, Y. Horiuchi, and S. Kuroiwa, "Text-independent speaker identification based on feature transformation to phoneme-independent subspace," in *Int. Conf. Communication Technology (ICCT)*, 2008, pp. 692–695.

[78] X.-C. Lu, J.-X. Yin, and W.-P. Hu, "A text-independent speaker recognition system based on probabilistic principle component analysis," in *Int. Conf. on System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, 2012, vol. 1, pp. 255–260.

[79] J. Wang, A. Ji, and M. T. Johnson, "Features for phoneme independent speaker identification," in *Int. Conf. Audio, Language, Image Processing (ICALIP)*, 2012, pp. 1141–1145.

[80] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman, "MLLR transforms as features in speaker recognition," in *Proc. of the 9th European Conference on Speech Communication and Technology*, 2005, pp. 2425–2428.

[81] M. Ferras, C.-C. Leung, C. Barras, and J. Gauvain, "Constrained MLLR for speaker recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2007, vol. 4.

[82] A. Stolcke, A. Mandal, and E. Shriberg, "Speaker recognition with region-constrained MLLR transforms," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2012, pp. 4397–4400.

[83] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[84] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the mllr framework," *Computer Speech and Language*, vol. 10, pp. 249–264, 1996.

[85] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *IEEE Trans. Speech, Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

[86] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK book*, 2006.