AUTOMATIC SPEAKER RECOGNITION AND DIARIZATION

IN CO-CHANNEL SPEECH

by

Navid Shokouhi

APPROVED BY SUPERVISORY COMMITTEE:

John H. L. Hansen, Chair

Carlos Busso

Issa M. S. Panahi

P. K. Rajasekaran

Copyright © 2017

Navid Shokouhi

All rights reserved

*Dedicated to my parents, Hossein and Manzar,*

*and my brother, Ali*

# AUTOMATIC SPEAKER RECOGNITION AND DIARIZATION IN CO-CHANNEL SPEECH

by

NAVID SHOKOUHI, BS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

May 2017

# ACKNOWLEDGMENTS

This study would not have been possible without the support of my advisor, Professor John H. L. Hansen. For years he has managed the Center for Robust Speech Systems (CRSS), where several students and research staff have benefited from his professional and scientific support. I would also like to acknowledge all the students and staff at CRSS whose presence was an encouragement for remaining productive during the course of this study.

I thank Dr. Carlos Busso, Dr. Issa Panahi, and Dr. P. K. Rajasekaran for agreeing to sit as committee members and assess my work. I would especially like to thank Dr. Rajasekaran, who in addition to being a committee member, provided constant advice and encouragement during my years as a PhD student.

At last I thank my parents, Hossein Shokouhi and Manzar Mohammadi. My father, Hossein, functioned as a second PhD advisor for me from across the globe by sharing his own experience as a PhD student and encouraging me throughout my work.

This project was funded by AFRL and partially by The University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by Professor John H. L. Hansen.

November 2016

AUTOMATIC SPEAKER RECOGNITION AND DIARIZATION

IN CO-CHANNEL SPEECH

Navid Shokouhi, PhD
The University of Texas at Dallas, 2017

Supervising Professor: John H. L. Hansen, Chair

This study investigates various aspects of multi-speaker interference and its impact on speaker recognition. Single-channel multi-speaker speech signals (aka co-channel speech) comprise a significant portion of speech processing data. Examples of co-channel signals are recordings from multiple speakers in meetings, conversations, debates, etc. The nuisances of co-channel speech are two-fold: 1) overlapped speech, and 2) non-overlapping speaker interference. In overlap, the direct effects of two stochastically similar, non-stationary signals added together disrupts speech processing performance, originally developed for single-speaker audio. For example, in speaker recognition, identifying speakers in overlapped segments is more difficult compared to single-speaker signals. Analyses in this study show that introducing overlapped speech increases speaker recognition error rates by an order of magnitude. In addition to the direct impact of overlap, its secondary effect is in how one speaker forces the other to change his/her speech characteristics. Different forms of co-channel data are investigated in this study. In scenarios where the focus is on overlap, independent cross-talk is used. Independent cross-talk refers to the summation of independent audio signals from different speakers to simulate overlap. The alternative form of data used in this study is real conversation recordings. Although conversations contain both overlapped and non-overlapped speech, independent cross-talk is a better source of overlap. The reason real con-

versations are not deemed sufficient for overlap analysis is the scarcity and non-uniformity of overlaps in typical conversations. Independent cross-talk is obtained from the GRID corpus, which was used in the speech separation challenge as a source of overlapped speech. Real conversations are obtained from the Switchboard telephone conversation corpus. Other real conversational data used throughout this study include: the AMI meeting corpus, Prof-life-log, and UTDrive data. These datasets provide a perspective towards environment noise and co-channel interference in day-to-day speech. Categorizing datasets allows for a meticulous analysis of different aspects of co-channel speech. Most of the focus in analyzing overlaps is presented in the form of overlap detection techniques. This study proposes two overlap detection methods: 1) Pyknogram-based 2) Gammatone sub-band frequency modulation (GSFM). Both methods take advantage of the harmonic structure of speech to detect overlaps. Pyknograms do so by enhancing speech harmonics and evaluating dynamics across time, while GSFM magnifies the presence of multiple harmonics in different sub-bands. The other advancements proposed in this study use back-end modeling techniques to compensate for co-channel speech in real conversational data. These techniques are presented to reduce the impact of interfering speech in speaker-dependent models. Several methods are investigated, all of which propose a different modification to the popular probabilistic linear discriminant analysis (PLDA) used in state-of-the-art speaker recognition systems. In addition to model compensation techniques, this study presents CRSS-SpkrDiar, which is a speaker diarization research platform aimed at tackling conversational co-channel speech data. CRSS-SpkrDiar was developed during this study to alleviate end-to-end co-channel speech analysis. Taken collectively, the speech analysis, proposed features, and algorithmic advancements developed in this study all contribute to an improved understanding and measurable performance gain in speech/speaker technology for the co-channel speech problem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

# CHAPTER 1

# BACKGROUND

The presence of speech interference is a challenging issue for all automatic speech processing systems. As speech technology advances into our daily lives (in the form of text-to-speech recognition, speaker verification systems, etc.), the need to address multi-speaker interference increases. This increase is due to the number and amount of naturalistic human-to-human interactions captured for archiving, entertainment, social media, etc. In this study, multi-speaker interference is referred to as co-channel speech. Precisely, *co-channel speech* refers to single-channel audio data that contain more than one speaker. Channel, in our definition, is synonymous to *recording device*; hence single-channel implies access to only one recording. Unfortunately, due to the non-stationary nature of speech interference, co-channel speech is inherently a difficult type of signal for speech processing. The presence of multiple speakers in co-channel increases the complexity of problems that are already difficult to address in single speaker scenarios. However, advancements in single-speaker speech technology over the past two decades has allowed researchers to broaden their scope of interest. Today, automatic speech recognition (ASR) is enabled in most smart-phones. During the course of this study, both Microsoft Windows and Mac OS were released with built-in speech recognition capability and one comes with a voice-based user verification system. These signs reflect the reality that current speech technology is extending its reach into our daily activities. A significant portion of day-to-day speech data captured by devices is co-channel. This investigation is therefore partly due to the rise of a new age in speech technology where research is not limited to isolated single-speaker conditions.

The focus of this study is to address various aspects of speaker recognition in co-channel speech data. This is accomplished by providing a clear definition of co-channel speech and its various forms. We will see that not only different solutions, but different approaches must be used to address each form of co-channel.

A wide range of terms have been used to describe various aspects of co-channel speech, which will be clarified throughout this chapter. This study addresses both conversational speech and artificially mixed audio streams as co-channel. Although some studies propose multi-channel solutions to co-channel speech (Panahi and Venkat, 2009; Xiao et al., 2011), the focus here is solely on single-channel recordings (i.e., a single microphone). In co-channel data, a subset of instances may contain more than one "active" speaker at the same time (i.e., multi-speaker segments), which we label as "overlapped speech". Overlapped regions are segments of a co-channel signal where both speakers are simultaneously active. This categorization is summarized in Fig. 1.1.

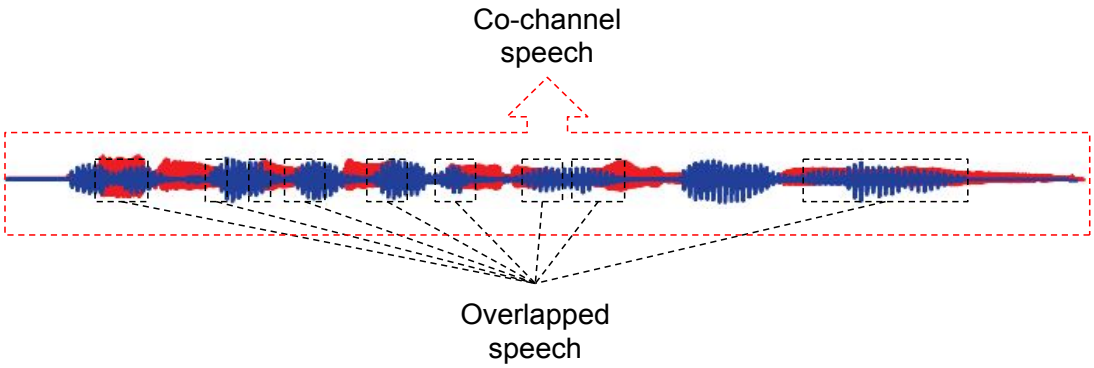


Figure 1.1. Difference between co-channel and overlapped speech. Overlap refers to instances where more than one speaker is active. Co-channel is defined as an entire stream that contains multiple speakers. All co-channel files do not necessarily contain overlap.

Aside from overlapped vs. non-overlapped speech, there are different ways of categorizing co-channel data in terms of how it is generated. Among such is a semantic classification that focuses on speakers' interactions with each other and divides co-channel data into two subgroups:

- conversational co-channel speech
- co-channel speech with independent parties (i.e., independent cross-talk)

Conversational co-channel speech refers to recordings in which speakers acknowledge other parties in the recording and engage in dialog. Alterations of speech production are

an important artifact of conversational co-channel speech, which are the result of conscious and unconscious reactions of the foreground speaker and interferer(s). Examples of such alterations are raised pitch and energy level (Shriberg et al., 2001; Schegloff, 2000). Raised Pitch and volume are especially common at or around overlaps. Readers are probably familiar with political debates with heated arguments. Most debates are perfect instances of an exaggerated version of the above-mentioned changes in speech production. Schegloff argues that long and sustained overlaps are primarily a sign of argumentative speech (Schegloff, 2000). In such conversations, most speakers tend to alter their voice in order to control the floor. These changes are problematic in automatic speech applications and are considered a type of distortion. Treatments are directed towards applications that suffer the most from speech alterations, predominantly automatic transcription of speech, i.e., speech recognition. Aside from changes in speech production, the element of interference by competing speakers is also an important artifact observed during overlaps. Therefore, in conversational co-channel data, one has to consider both overlaps and speaker specific alterations as sources of distortion and mismatch.

Co-channel data with independent parties, are examples of co-channel data where the speakers do not interact with each other. An example of this type is cross-talk between separate channels; imagine switching between radio stations on an analog AM radio. The main characteristic of such data is that speakers are not aware of each other and therefore do not pertain to normal conversational mannerisms. That includes following turn-taking rules, which in most cases limits overlapped speech. Artificially generated co-channel data (mixing independent channels) is another example of independent cross-talk. A considerable portion of this study will focus on this type of co-channel data to analyze performance of overlap detection and also speaker recognition. We rely on this type of data since it provides the flexibility of controlling the amount of overlapped speech. As we will show in the next chapter, conversational co-channel speech does not necessarily contain sufficient

overlapped data for some of our experiments. Therefore, we allow ourselves to neglect some aspects of conversational co-channel speech for the benefit of more overlap. This is the main motivation to value "co-channel data with independent parties", despite some of its unrealistic characteristics compared to natural conversational speech.

The treatment of different speech processing applications for co-channel speech will focus on one of the two categories described above. This study will focus on a number of speech applications including:

- **Signal processing and audio classification:** identifying and separating overlapped segments in co-channel files.
- **Speaker recognition/verification:** The ability to automatically decide whether two or more speech samples belong to the same speaker.
- **Speaker diarization**: Segmenting an audio stream by counting the number of speakers as well as determining who spoke when.

A description of an outreach to signal processing in vehicles is also presented, where algorithms developed for co-channel speech analysis were used to improve driver safety.

### 1.1 Approach

The goal of this thesis is to provide tangible solutions to problems caused by co-channel speech in automatic speaker recognition. We suggest that part of these issues are caused by overlapped speech (direct speech interference), which plays a significant role in making co-channel speech a difficult problem. The presence of overlapped speech can be detrimental to both speaker diarization and speech recognition systems. In speaker diarization, it becomes difficult to assess system performance during overlaps, due to the inherent ambiguity in labeling overlapped segments. The same goes for speech recognition where aside from

determining which is the "primary" speaker, recognizing speech at overlaps is more difficult, due to interference stemming from other speakers. For this and other reasons detailed in the next chapter, the first portion of this study is devoted to overlapped speech detection. Our approach to overlap detection will be to focus on developing signal processing techniques to detect and separate overlap from single-speaker speech. Figure 1.2 depicts some applications of overlap detection in speech technology.

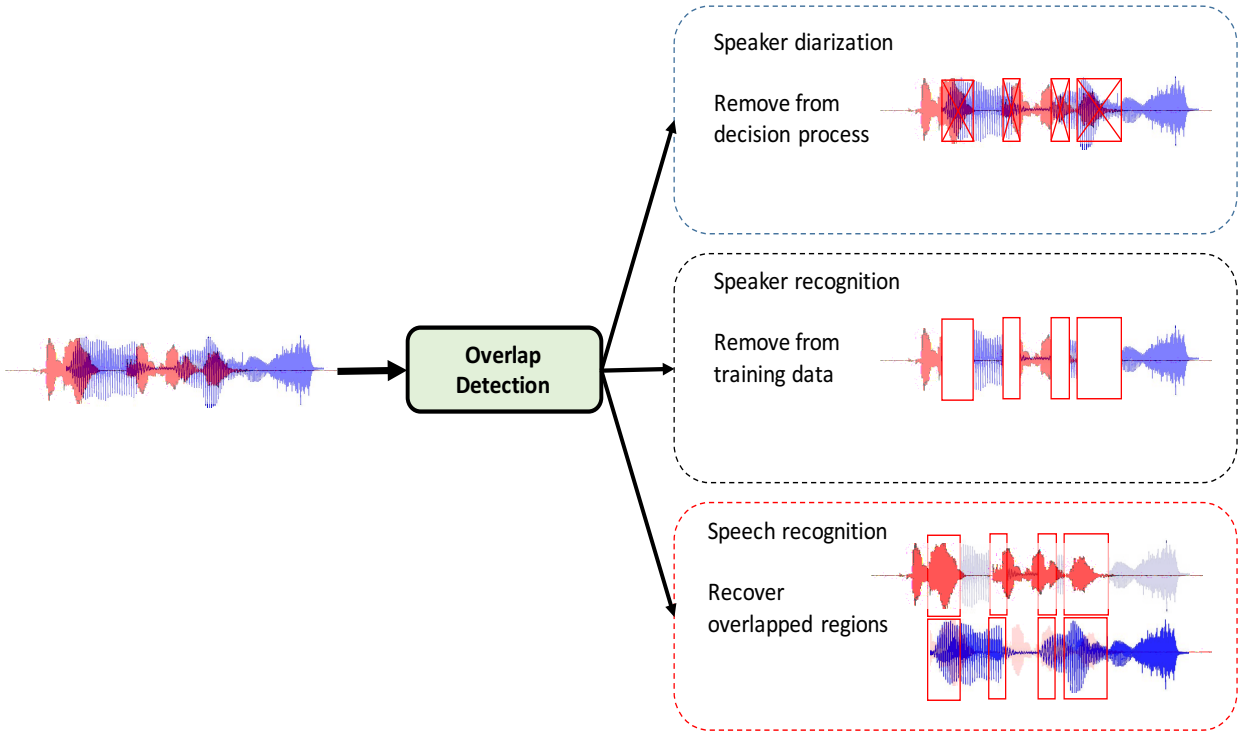


Figure 1.2. Various applications of overlap detection. Top: In speaker diarization, ignoring overlapped regions provides a more fair assessment of diarization performance. Middle: Removing overlaps in speaker recognition increases the reliability of training data. Bottom: Overlap detection results can be used as an initial step to recover overlapped regions.

Although overlap is considered an important aspect of co-channel speech, for many conversational speech data resources, the amount of overlap can be considered negligible, depending on the application. One such application is speaker recognition, which is also the main theme of this thesis. In text-independent speaker recognition, speech content (i.e.,

what is being said) is of less values compared to long-term speaker dependent characteristics (i.e., who is speaking). The standard approach in dealing with co-channel speech in such cases is speaker diarization. The role of diarization is to segment audio streams into shorter intervals each of which contains only one speaker. We consider speaker diarization to be a "signal level" approach. The term signal level is used with respect to co-channel speech, since it is removed from the original signal prior to any subsequent processing. Alternately, a novel approach is presented in this study with the intention of bypassing the use of speaker diarization in the aforementioned scenario while preserving speaker recognition performance. This approach is to remove unwanted speaker-dependent information from latent variable subspaces generated from the audio files using the i-vector framework established by (Dehak et al., 2011; Kenny, 2010). We refer to such solutions as "subspace level". Therefore, Chapters 4 and 5 each propose a different approach to remove interfering speakers from co-channel data.

1. Remove interfering speakers at the feature subspace level: i-vector subspace factorization.

2. Remove interfering speakers at the signal level: speaker diarization.

Speaker diarization, will attempt to recognize and group speech that belongs to the same speaker in a co-channel audio stream, while subspace factorization maps speaker-dependent models to a subspace that will only contain parameters identifying the speaker of interest (aka primary speaker). Once again it should be noted that the main theme of this study is to address speaker recognition and identification in co-channel speech. Which makes speaker diarization an inseparable part of our approach.

## 1.2 Outline

This dissertation is organized in the following manner. Thesis contributions are split into two main categories: 1) signal processing approaches, and 2) probabilistic modeling. Chapter 2,

overlap detection methods, presents signal processing techniques used for overlap detection. Chapter 3, speaker recognition in overlapped speech, highlights the importance of addressing overlapped speech in speaker recognition problems. In addition, techniques proposed Chapter 2 are used in Chapter 3 to improve recognition performance. Chapter 4, speaker recognition in co-channel speech, focuses on probabilistic modeling techniques to improve speaker recognition in co-channel.

In addition to the two primary contributions, this thesis also covers practical aspects of co-channel speech processing. These aspects are covered in two frameworks. The first is the description of CRSS-SpkrDiar (Chapter 5), which is an end-to-end speaker diarization system. The second aspect is an exhibition of interdisciplinary applications of co-channel speech in other signal processing applications (Chapter 6).

Finally, a summary of conclusions and references to future studies are presented in Chapter 7.

### 1.3 Dissertation Contributions

This dissertation investigates various aspects of co-channel speech and is novel in a number of ways. The first important contribution is to:

- separate overlapped speech from co-channel.

The significance of this contribution is in its alternative perspective compared to existing studies. Solely focusing on overlap instead of co-channel (or vise versa) limits the applicability of methods and makes it difficult to compare studies, since different studies refer to different phenomena as co-channel.

The second contribution is from a signal processing perspective, which is to:

- propose features to detect overlapped speech from single-speaker speech (overlapped speech detection).

This is important, since overlap detection constitutes a significant portion of co-channel research. Being able to detect overlapped segments is also useful in other problems related to co-channel analysis. One of the applications of overlap detection in co-channel speech is used in speaker diarization systems. The contribution in regards to speaker diarization is to:

- provide a detailed description of CRSS-SpkrDiar, a speaker diarization tool-kit.

The last contribution of this study focuses on latent space analysis, which is to:

- develop and analyze modified probabilistic linear discriminant analysis to suppress interfering speech to improve speaker recognition performance.

The theme in all of the aforementioned contributions is speaker recognition. Therefore, all techniques and analyses provided in the next chapters are in accordance with speaker recognition frameworks and are intended to improve speaker recognition performance.

Table 1.1. LIST OF ACRONYMS

| | |
|---|---|
| API | Application Programming Interface |
| ASR | Automatic Speech Recognition |
| BIC | Bayesian Information Criterion |
| DCF | Detection Cost Function |
| DER | Diarization Error Rate |
| DESA | Discrete Energy Separation Algorithm |
| EER | Equal Error Rate |
| EM | Expectation Maximization |
| ERB | Equivalent Rectangular Bandwidth |
| FA | False Alarm |
| FM | Frequency Modulation |
| GLPK | GNU Linear Programming Kit |
| GMM | Gaussian Mixture Model |
| GSFM | Gammatone Sub-band Frequency Modulation |
| HMM | Hidden Markov Model |
| ILP | Integer Linear Programming |
| LDA | Linear Discriminant Analysis |

MFCC ..... Mel Frequency Cepstral Coefficients

NIST ..... National Institute of Standards and Technology

OVL ..... Overlap

PLDA ..... Probabilistic Linear Discriminant Analysis

RTTM ..... Rich Transcription Time Marks

SAD ..... Speech Activity Detection

SAPVR ..... Spectral Autocorrelation Peak-Valley Ratio

SFM ..... Spectral Flatness Measure

SIR ..... Signal-to-Interference Ratio

SNR ..... Signal-to-Noise Ratio

SRE ..... Speaker Recognition Evaluation

SSC ..... Speech Separation Challenge

SVM ..... Support Vector Machine

TEO ..... Teager Energy Operator

TV ..... Total Variability

UBM ..... Universal Background Model

WCE ..... Word-Count Estimation

# CHAPTER 2

# OVERLAP DETECTION METHODS[1]

Overlapped speech constitutes a significant amount of research in co-channel speech. To such an extant that in many cases the terms co-channel and overlapped speech are used interchangeably. Single-channel recordings from meetings or conversations, which we defined as co-channel speech in Chapter 1, are examples of signals during which speakers may overlap. This chapter focuses on proposing signal processing techniques to recognize instances of co-channel data where speakers overlap (i.e., overlapped speech detection). Therefore, we are interested in identifying only those portions of co-channel speech that are overlapped.

Section 2.1 provides preliminary information required to understand the focus of this chapter. This section states the significance of overlap detection over speech enhancement techniques in the context of speaker recognition and diarization problems. Section 2.2 presents a history of overlap detection methods and the various approaches used to distinguish single-speaker speech from overlap. Section 2.3 introduces the GRID dataset, which is used throughout the chapter. GRID focuses on independent cross-talk (see Chapter 1), which provides sufficient overlap with variable/controllable interference levels. Sections 2.4 and 2.5 propose two overlap detection techniques used to identify overlap from single-speaker speech. Section 2.6 points out a significant drawback of GSFM in real-noise conditions and shows how Pyknograms are superior in that regard.

The primary contributions of this chapter are two proposed overlap detection algorithms:

- Pyknogram-based overlap detection
- Gammatone Sub-band Frequency Modulation features

[1]Portions of this chapter were adopted from published material with the authors' full consent. Shokouhi, Navid, Ali Ziaei, Abhijeet Sangwan, and John H. L. Hansen, "Robust overlapped speech detection and its application in word-count estimation for Prof-Life-Log data," In Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP) ©2015 IEEE.

## 2.1 Why overlap detection?

In contrast to what overlap detection entails, most studies on overlapped speech have focused on separating speech from the target speaker or suppressing interfering speech (Morgan et al., 1997). Such enhancement techniques are often used to de-noise and thereby improve the performance of automatic speech applications (Quatieri and Danisewicz, 1990; Chazan et al., 1993; Cooke et al., 2010) (primarily speech recognition). However, due to increased interest in recognition systems such as speaker recognition and diarization, a growing trend of detecting overlapped regions has been observed. In speaker recognition, the presence of interfering speech in conversational speech styles not only reduces the effectiveness of trained speaker models but also increases the uncertainty in scoring test files with overlapped regions (Yantorno, 1999). Removing overlapped segments increases model reliabilities to improve recognition performance (Shokouhi and Hansen, 2015). State-of-the-art speaker diarization systems are also currently at a stage where one of the main sources of error is the presence of overlapped speech (Boakye et al., 2008; Zelenak et al., 2012). One of the reasons overlaps become a source of confusion in speaker diarization systems is that there is no basis for selecting ground-truth in overlapped regions. This makes evaluating speaker diarization systems more challenging.[2] Fortunately, for speaker recognition and diarization it is rarely necessary to separate the target from interfering speaker in overlapped speech, since speaker identities are considered long-term features and short-term features (phones, words, etc.) are less valuable.

One can improve system performance by detecting and excluding overlapped segments. In other words, removing a corrupt (in this case overlapped) speech segment usually does more good then harm in such applications. Replacing interferer suppression and target separation with overlapped speech detection, is sometimes called "usable speech detection" (Yantorno,

[2]Future chapters will describe co-channel speech data in speaker diarization in more detail.

1999).[3] An overlapped speech detection system can be used in any of the aforementioned tasks as a data purification step or a signal processing front-end. Signal processing front-end solutions in our study focus solely on overlapped speech detection. Figure 2.1 summarizes incorporating overlap detection in speech processing technology.

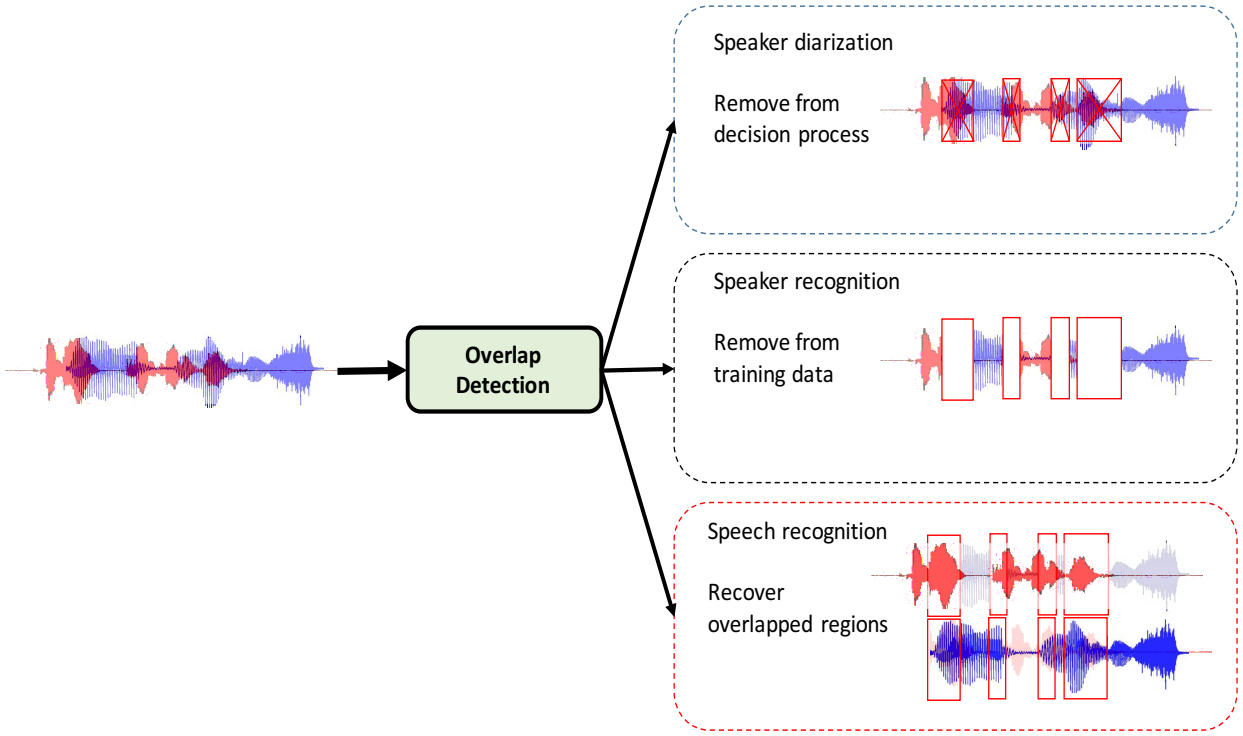


Figure 2.1. Applications of overlap detection. Top: In speaker diarization, removing ignoring overlapped regions provides a more fair assessment of diarization performance. Middle: Removing overlaps from in speaker recognition increases the reliability of training data. Bottom: Overlap detection results can be used as an initial step to recover overlapped regions. ©2017 IEEE

[3]In order to avoid any confusion between this study and the assumptions made in (Yantorno, 1999), we use the more general term overlapped speech detection.

## 2.2 Background

Traditionally, studies have used spectral harmonicity as a key factor in detecting overlapped speech (Shokouhi et al., 2013; Smolenski and Ramachandran, 2011). This approach is motivated by the fact that two fundamental frequencies exist in most instances of overlapped speech which disarranges the harmonic structure observed in single-speaker speech. As a side-note here, we point out that most of the focus in overlapped speech has been on regions where both speakers produce "voiced" speech. In (Morgan et al., 1997) a classification of different types of segments in co-channel speech was presented.

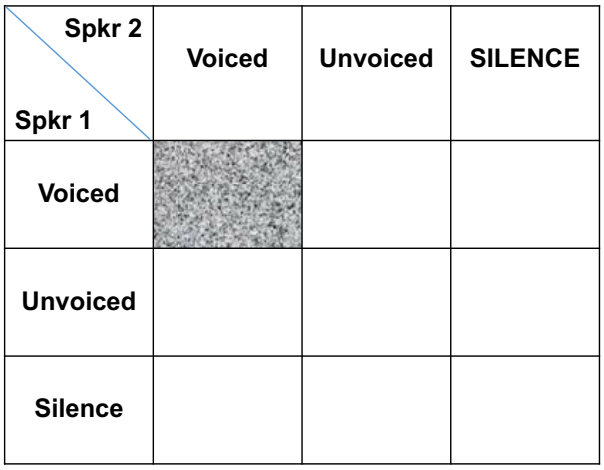


Figure 2.2. Classification of different segments in a co-channel file. In overlap detection we are interected in the voiced-voiced (shaded) region.

Most of the attention in this study will also be on the voiced-voiced cell, since detecting other regions is far more difficult and voiced segments are more important for speaker recognition, which is the theme of this thesis. A more detailed classification of overlapped regions is presented in (Shokouhi et al., 2013), where a grid containing all phones is used to rank-order overlapped segments in terms of difficulty. The analysis in (Shokouhi et al., 2013) expands Fig. 2.2 as shown in Fig. 2.3. The general consensus, however, is to focus on detecting voiced-voiced overlap detection, which from now on we will refer to as overlap detection.

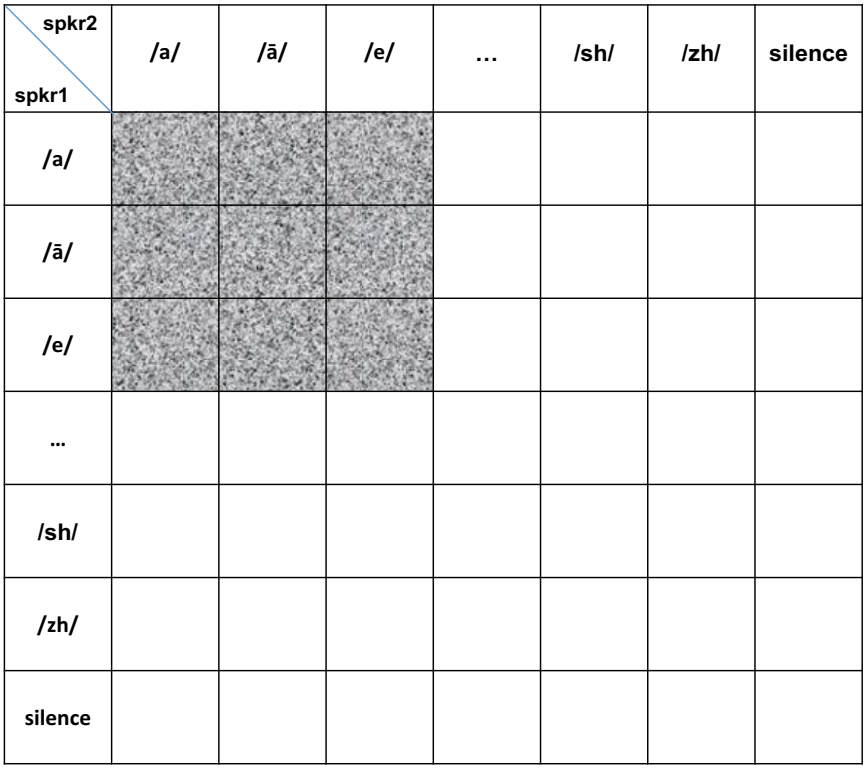


Figure 2.3. phone-based expansion of overlapped segments in Fig. 2.2.

Concentrating on voiced speech yields more discriminating harmonic structures. Essentially, the defining factor here is that single-speaker voiced spectra are represented as harmonics of a single fundamental frequency. The presence of overlapping voiced speech introduces harmonics of a second fundamental frequency, such secondary harmonics are unlikely to be exactly aligned with those of the first speaker. In (Krishnamachari et al., 2000), the peak-to-valley ratios in frame-based spectral autocorrelations are introduced as a discriminating feature for overlapped speech detection through the same assumption. Spectral flatness measure (SFM), the ratio of geometric to arithmetic means calculated from spectral bins in a speech frame, has also been used as a measure to capture harmonicity and has been used to detect the presence of overlapped speech (Shokouhi et al., 2013). Another related characteristic is observed when monitoring fundamental frequencies along time. Adjacent pitch period comparison (APPC) presented in (Lovekin et al., 2001) uses the temporal vari-

ation of estimated "pitch" periods as a measure to detect "usable" speech with the assumption that temporal variations of adjacent pitch periods are significantly higher in overlap. A multi-pitch tracking algorithm proposed in (Wu et al., 2003) was used in (Shao and Wang, 2003) to estimate coexisting fundamental frequencies in the presence of multiple speakers. Regions where more than one fundamental frequency is estimated are labeled as overlap. The multi-pitch tracking technique described in (Wu et al., 2003), decomposes speech into sub-bands and pitch estimation is only performed on reliable sub-bands.

A slightly different, yet fundamentally similar, approach to distinguish overlapped speech is to use the speech kurtosis which measures higher order moments of the signal statistics (Wrigley et al., 2005).

A number of studies have also considered investigating spectral characteristics at formant frequency locations when dealing with overlapped speech. Giuliani et al. use a filter-based approach to improve speech recognition rates for different instances of meeting conditions by adding a detection step that separates double-speaker speech from single-speaker audio (Giuliani et al., 2006). This was accomplished by cascading two-layer sub-band filters to capture formant characteristics. Formant frequency information was obtained by filtering the signal at sub-bands with center frequencies and bandwidths corresponding to nominal $F_1, F_2$, and $F_3$ values for all English vowels. One of the reasons Formant-based overlapped speech analysis has received less attention is the difficulties in modeling pole interactions at overlapped regions, which is an issue for linear predictive modeling and other commonly used formant tracking techniques. Characterizing pole interactions using standard LP models easily becomes intractable in the presence of more than one source. Add to this complication, the unknown speaker locations with respect to each other and the microphone. As a result, we focus our attention to nonlinear speech models, some of which have been proven to be more successful in the scenarios described above.

Nonlinear speech models, including the AM-FM speech model (Maragos et al., 1993) have been used in previous studies to model speech resonances without any specific requirements

for the source signal. These energy operators have also been used to deal with signals with more than one source (Maragos et al., 1995), aka co-channels signals.[4] Maragos et al. used higher order energy operators to develop an algorithm that simultaneously demodulates the components of a co-channel mixture in AM-FM modulated signals (Maragos et al., 1995). Litvina et al. separated speech from music using the Teager energy operator (TEO) separation algorithm (Maragos et al., 1993) (Litvin et al., 2010), where they used the extracted components to design a time-varying filter and suppress the interfering signal. Similar multicomponent signal decomposition techniques have been addressed using energy operators to separate narrow-band signals (Lin et al., 1995; Hu et al., 2012; Santhanam and Maragos, 2000).

Our goal is to incorporate sub-band analysis to design a technique suitable for **overlapped speech detection**. Two algorithms are proposed that incorporate sub-band analysis for overlap detection:

- using Teager-Kaiser energy operator (TEO) methods on narrow-band components to detect single- vs. double-speaker speech harmonics,

- apply cosine functions across sub-band outputs to magnify the presence of multiple harmonics.

### 2.2.1 Baseline features

Below, a collection of overlap detection features are presented that have previously been used to detect overlapped regions (Shokouhi et al., 2013; Boakye, 2008; Krishnamachari et al., 2000). To the best of our knowledge, overlap detection results on the GRID database

[4]Co-channel is a more general terminology used to described multi-component signals. In the case of speech, co-channel speech may refer to any single-channel recording that contains speech from multiple speakers, regardless of whether there is overlap.

(see Sect. 2.3) have not been reported for any of the following features, therefore the only reference for comparison are our in-house implementations.

- *Speech kurtosis*: Kurtosis has been reported as an effective measure to detect the presence of multiple speakers in overlapped signals by several studies (Wrigley et al., 2005; Boakye, 2008; Krishnamachari et al., 2001). It has been shown that overlapped speech exhibits lower kurtosis compared to single-speaker speech (LeBlanc and De Leon, 1998). The kurtosis of a zero-mean random variable $x$ is defined as:

$$k_x = \frac{E\{x^4\}}{(E\{x^2\})^2}. \tag{2.1}$$

  In this case, $x$ refers to speech samples in a given frame.

- *Spectral flatness measure (SFM)*: The ratio of the geometric to arithmetic means of spectral magnitudes across frequency within each frame (Shokouhi et al., 2013). For the $i^{th}$ frame:

$$sfm_i = \frac{\frac{1}{N}\sum_{n=1}^{N} X(f_n)}{\sqrt[N]{\prod_{n=1}^{N} X(f_n)}}, \tag{2.2}$$

  where $X(f_n)$ corresponds to the magnitude spectrum at frequency $f_n$ and N is the total number of frequency bins.

- *Spectral autocorrelation peak-valley ratio (SAPVR)*: described briefly in the beginning of the section, this feature uses the dominance of peaks in the spectral autocorrelation in each frame as a measure to detect overlaps (Krishnamachari et al., 2000).

## 2.3 Data: Monaural Speech Separation Challenge

Before moving forward, a description of the data used in the monaural speech separation challenge is provided. This data is used throughout the chapter in the analysis of overlapped speech. Since the prime focus here is overlapped speech and its physical attributes, we rely on independent cross-talk data for the experiments (see definition in Chapter 1).

The data used in our controlled experiments is from the monaural speech separation and recognition challenge (aka speech separation challenge (SSC)) (Cooke et al., 2010). The objective in the SSC was to permit a large-scale comparison of techniques for the overlapped speech problem (Cooke et al., 2010). Participants were asked to identify keywords in sentences spoken by a target talker when mixed into a single channel with a background talker speaking sentences of the same structure but with different content. The data used in SSC was obtained from the larger GRID corpus (Cooke et al., 2006), which is a multi-talker audio-visual sentence corpus that supports computational-behavioral studies in speech perception. In this study, only the audio content is used. The audio consists of 1000 sentences spoken by each of 34 talkers (18 male, 16 female). The sentences are structured in the following format.

**<command><color><preposition><letter><number><code>**

For example, "lay white at X six now".

Seven overlapped sets are available, one clean and the rest are composed of sentence pairs that are artificially summed at 6 signal-to-interference ratios (SIR) (+6, +3, 0, -3, -6, -9 dB). Since file durations are short (typically less than 5 seconds) and the utterances contain negligible pauses, it is reasonable to consider the average SIR values, provided for each file, a fair representation of the amount of overlap. It is also safe to assume that a given file is all speech, removing the need to run speech activity detection to separate speech from silence. This assumption justifies labeling a "clean" file (no overlap) as single-speaker.

For overlaps, it is reasonable to consider the entire co-channel signals to be overlapped (see Fig. 2.4). All files have been down-sampled to 8kHz to match telephone recordings. Note that the experiments conducted in this study do not comply with the objectives of the speech separation challenge described in (Cooke and Lee, Cooke and Lee).

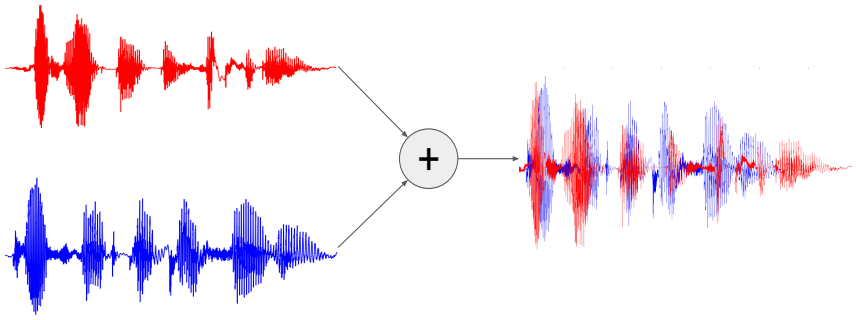

Figure 2.4. Example of the mixing process for a 0dB SIR overlapped signal. As shown on the right, it is fair to assume that overlap occurs throughout the signal.

The choice of dataset for this study is pertinent to the investigative objective of this chapter, overlap detection. Other studies have also focused on controlled datasets such as TIMIT (Shokouhi et al., 2013) as well as realistic datasets including Prof-Life-Log (PLL) (Ziaei et al., 2013; Shokouhi et al., 2015), Switchboard (Shokouhi and Hansen, 2015), and UT-Drive in-vehicle conversations (Sathyanarayana et al., 2013). Some of these studies are described in following chapters. Unfortunately, none of the aforementioned datasets are designed to contain significant amounts of overlap. One may argue that overlaps exist in conversational speech corpora such as Switchboard or the AMI meeting corpus (Carletta et al., 2006). Although it is true that these corpora contain overlapped segments, the amount of overlap in "regular" (i.e., non-competitive conversations (Schegloff, 2000)) is not sufficient for the requirements of this study. The reason control is required over the amount of overlap in experiments is to specifically investigate the power of proposed overlap detection systems and not worry about target/non-target imbalances observed in conversational speech. Furthermore, the GRID corpus is isolated from variabilities other than overlapped speech (e.g.,

microphone/headset mismatch), which makes it useful to study the effects of overlap. To the best of my knowledge, this dataset is the most organized publicly available corpus that contains large and controlled amounts of overlapped speech (note that we are mostly interested in *overlapped speech* and not *co-channel speech* as defined and distinguished in Chapter 1). Another advantage of the corpus is the fact that segments are short which makes the definition of a signal-to-interference ratio more appropriate. Had the signals been longer, say a few minutes long, the notion of a signal-to-interference ratio across the entire signal would have been less applicable, due to the non-stationary nature of speech.

Table 2.1 shows some properties of the dataset used in this study, including: number of speakers (male and female), average file duration, and overlap conditions.

Table 2.1. Summary of data used for speaker recognition experiments. ©2017 IEEE

| | |
|---|---|
| number of speakers | 18 (male)<br>16 (female) |
| average file duration | 1.9 (sec) |
| noise | interfering speakers<br>clean,+6, +3, 0, −3, −6, −9 dB |
| sampling rate | 8 KHz |

## 2.4 Pyknogram-based overlap detection

The first proposed overlap detection method is a novel approach for overlapped speech detection based on an enhanced spectrogram. These spectrograms, called Pyknograms, were first introduced by Potamianos and Maragos in (Potamianos and Maragos, 1995, 1996) and are calculated by applying multi-band demodulation in the AM-FM speech model framework (Maragos et al., 1993).[5] Pyknograms provide a more prominent representation of

[5]The authors in (Potamianos and Maragos, 1996) used the term "Pyknogram" which stems from the Greek word "pykno" meaning dense. Pyknograms represent highly resonating regions in time-frequency plots as populated scatter plots, hence the term density.

harmonic trajectories in a unique time-frequency space, which is proposed here to be used as a means to detect the presence of interfering speech.

Pyknogram extraction can be considered a 2 step process of obtaining a binary mask of time-frequency units for the amplitude spectrogram.

1. Frequency estimation: Computes resonant frequencies in time-frequency units. The procedure in this step includes:
    - apply a gammatone fiterbank to the speech signal
    - estimate instantaneous amplitude and frequencies using TEO
    - block the per sub-band outputs into time frames
2. Frequency selection: Prunes the estimated frequencies to find the most reliable units time-frequency units.

### 2.4.1 Pyknogram Extraction - Frequency estimation

In Pyknograms (Potamianos and Maragos, 1996), the harmonic structure of speech is enhanced by decomposing spectral sub-bands into amplitude and frequency components. This sub-band analysis uses the AM-FM speech model (Maragos et al., 1993) to decompose speech sub-bands and thereby calculate the corresponding instantaneous frequencies and bandwidths. To extract Pyknograms, the speech signal is initially passed through a filter-bank (the algorithm has been modified in this study to use logarithmically spaced Gamma-tone filters, while (Potamianos and Maragos, 1996) uses linearly-spaced Gabor filters). Filter-bank outputs ($x_i(n)$, in which $i$ represents filter indexes) are then decomposed into amplitude and frequency components using the discrete energy separation algorithm (DESA-1) (Maragos et al., 1993), where the per sample frequency of $x_i(n)$ is $f_i(n)$ and the amplitude is $a_i(n)$. Frequency and amplitude estimates for the $i^{th}$sub-band, $x_i(n)$, are:

$$f_i(n) = \frac{1}{2\pi} \arccos\Big(1 - \frac{\Psi[x_i(n) - x_i(n-1)]}{2\Psi[x_i(n)]}\Big), \tag{2.3}$$

$$|a_i(n)| = \sqrt{\frac{\Psi[x_i(n)]}{\sin^2(2\pi f_i(n))}}, \qquad i = 1, 2, ..., N_s \tag{2.4}$$

where $n$ is the time sample. $N_s$ is the number of sub-bands in the filter-bank and $\Psi(.)$ is the discrete energy operator (Kaiser, 1990, 1993), defined for any given signal, $x(n)$, in Eq. (2.5). Figures 2.6 and 2.7 show the amplitude and frequency estimates obtained using the DESA-1 algorithm for the input signal in Fig. 2.5.

$$\Psi[x(n)] = x^2(n) - x(n-1)x(n+1). \tag{2.5}$$

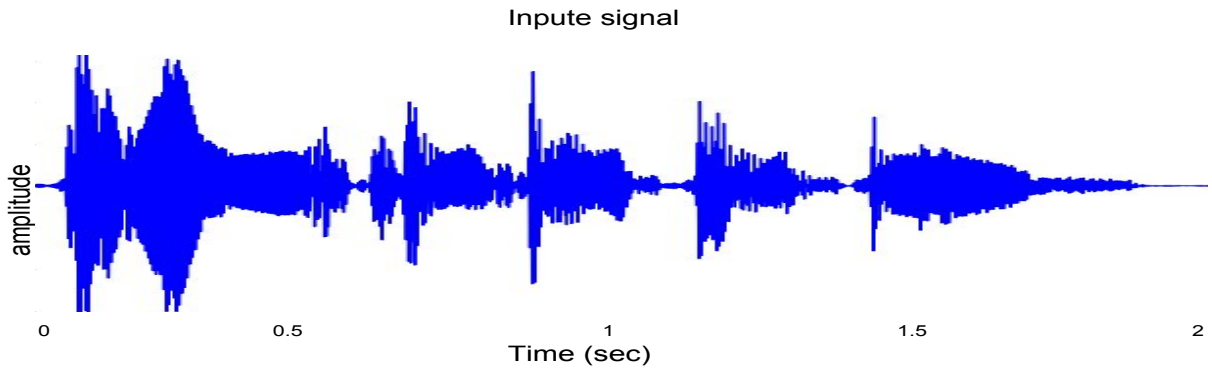


Figure 2.5. Input signal.

A weighted average of the instantaneous frequencies, $F_w$, is estimated over $25msec$ windows (i.e., frames), indexed by $t$. Together, sub-band analysis and time framing results in time-frequency units $(t, i)$, where $i$ corresponds to the frequency sub-band index and $t$ corresponds to frames (Cohen and Lee, 1990). Instantaneous frequencies are weighted using the estimated signal power ($|a_i(n)|^2$). The average frequency computed for each time-frequency unit can be viewed as the $1^{st}$-order moment of instantaneous frequencies.

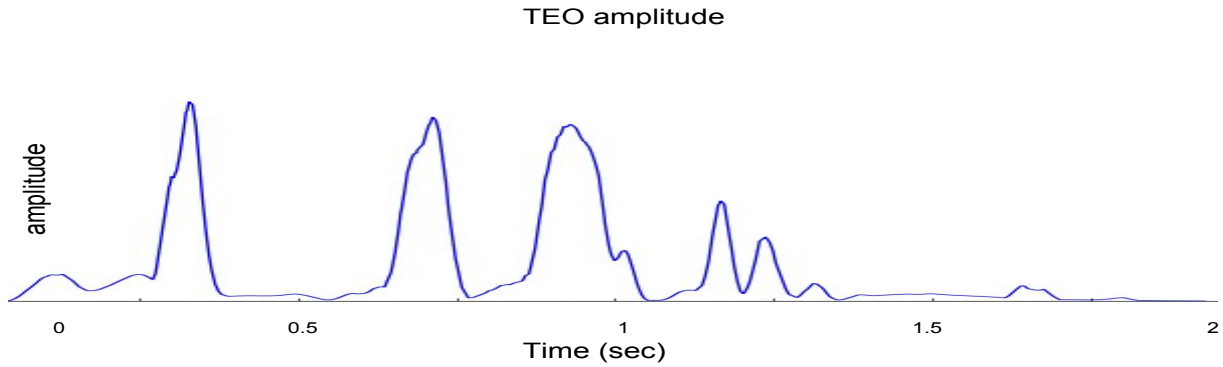


Figure 2.6. Outputs of DESA-1: Signal amplitude component estimated using TEO, Eq. (2.4). ©2017 IEEE

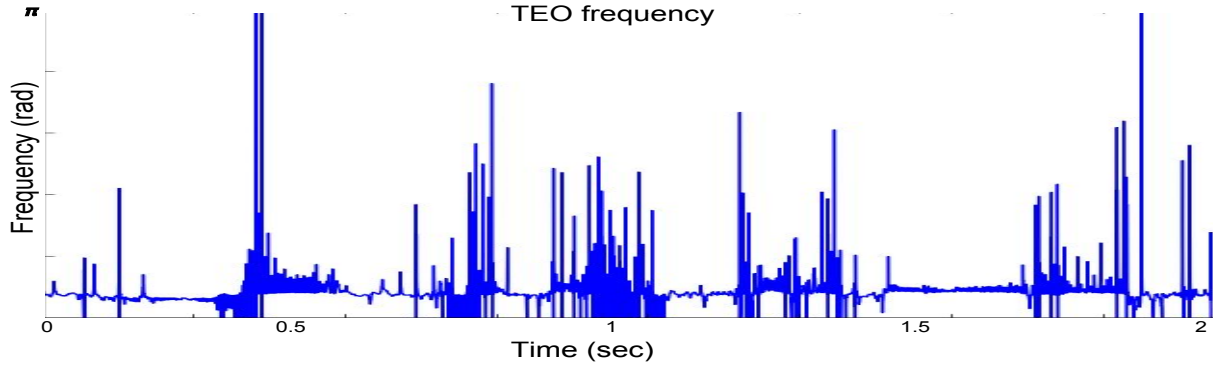


Figure 2.7. Outputs of DESA-1: Signal frequency component estimated using TEO, Eq. (2.3).

$$F_w(t,i) = \frac{\sum_{n_t}^{n_t+T-1} f_i(n) a_i^2(n)}{\sum_{n_t}^{n_t+T-1} a_i^2(n)}, \tag{2.6}$$

where $T$ is the number of samples per frame, from $n = n_t$ to $n = n_t + T - 1$, in which $n_t$ is the beginning sample of frame $t$. The algorithm also provides a way to estimate weighted bandwidths for the frequency component, Eq. (2.7). What are referred to here as bandwidths are essentially $2^{nd}$-order frequency moments.

$$B_w(t,i) = \sqrt{\frac{\sum_{n_t}^{n_t+T-1} (\dot{a}_i(n)/2\pi)^2 + (f_i(n) - F_w(t,i))^2 a_i^2(n)}{\sum_{n_t}^{n_t+T-1} a_i^2(n)}}, \tag{2.7}$$

where $f_i(n)$ and $a_i(n)$ are the instantaneous frequency and amplitude values from Eq. (2.3) and Eq. (2.4). In Eq. (2.6), the instantaneous frequencies are averaged over the $t^{th}$ frame using squared instantaneous amplitudes as weights. $\dot{a}(n)$ is the first difference of $a(n)$ (i.e., $a(n) - a(n-1)$). The per-frame values of $F_w$ provide initial estimates of spectrogram peaks. This results in a time-frequency, $t$-$f$, representation of the overall signal.

In (Potamianos and Maragos, 1996), the bandwidths, $B_w$, defined in Eq. (2.7) are used for analysis purposes. Here, they are used in overlap detection systems to determine the reliability of $t-f$ units. The assumption is that large Pyknogram bandwidths correspond to higher uncertainty in frequency estimates. This assumption is investigated in the following sections by adding an uncertainty term to frequency estimates proportional to the estimated bandwidth:

$$\tilde{F}_w(t,i) = F_w(t,i) + \epsilon_t, \tag{2.8}$$

where

$$\epsilon_t^i \sim \mathcal{N}(0, B_w(t,i)). \tag{2.9}$$

### 2.4.2 Pyknogram Extraction - Frequency selection

In the second step of Pyknogram extraction, dominant harmonic peaks are selected by comparing the average frequency estimates with the filter-bank center frequencies. According to (Potamianos and Maragos, 1996), points at which filter-bank center frequencies coincide with the weighted frequency estimates from Eq. (2.6) are more reliable in estimating spectrogram peaks. In Sect. 2.4.1, it was shown how resonant frequencies are estimated using Teager energy operators. A considerable number of these frequencies can be omitted from the list of candidate frequencies. The assumption here being that frequency estimates are more accurate when resonances align with a filter in the filter-bank. This defines the condition

through which initial $F_w$ values are tested to detect whether they correspond to prominent peaks. At frame $t$:

$$F_w(t,i) = F_c(i) \iff \{i \in peaks\}, \tag{2.10}$$

where $F_c(i)$ is the center frequency of the $i^{th}$ filter in the gammatone filter-bank. Note that center frequencies are distributed in a logarithmic scale. Another peak selection condition (as shown in Fig. 2.8) is to limit the relative variance of selected frequencies with respect to center frequencies.

$$\left|\frac{\partial F_w(t,i)}{\partial i}\right| \approx \left|\frac{F_w(t,i+1) - F_w(t,i)}{(i+1) - i}\right| < thr, \tag{2.11}$$

This condition limits non-harmonic anomalies that break the patterns in regular, single-speaker speech harmonics. Since such patterns are frequently observed in overlapped data, this restriction can be removed from the peak-picking step.

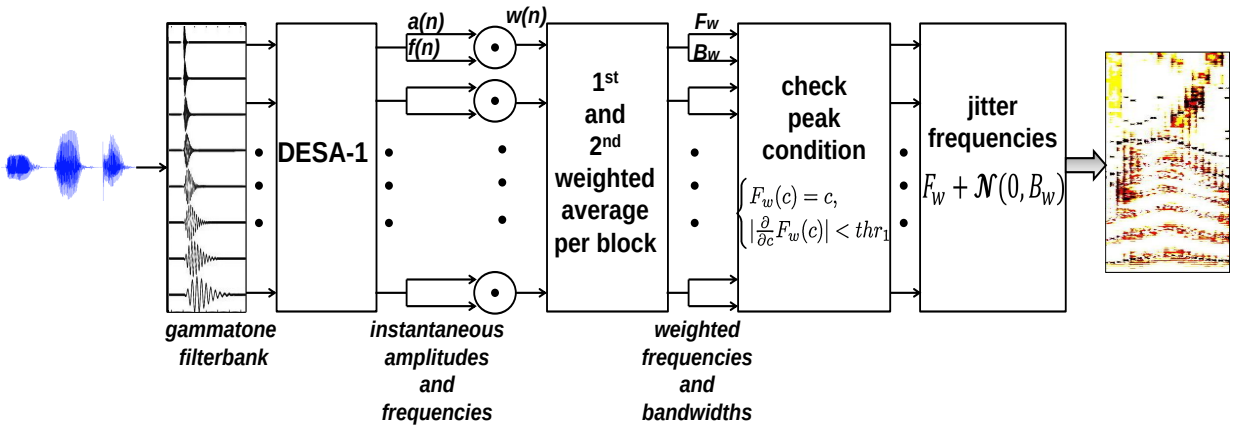


Figure 2.8. Pyknogram extraction block-diagram. ©2015 IEEE

One of the advantages of the peak-picking constraint in Eq. (2.10) is the quantization of spectrograms onto filter-bank center frequencies. This allows for the mapping of all signals onto a unified space defined by the filter-bank, which enables reliable comparison within the time-frequency space.

Using an energy operator based approach helps avoid assumptions on the number of speakers in the signal. AM-FM decomposition is suitable since it relies on signal resonances and does not restrict signals to a specific structure or number of speakers (as opposed to models such as linear prediction). The final time-frequency representation is called a Pyknogram, $S_{pyk}(t,i)$, which is a function of time ($t$) and frequency index ($i$). $S_{pyk}(t,i)$ is obtained by applying a binary mask to the gammatone time-frequency amplitudes estimated, $A(t,i)$, from Eq. (2.4) in the following manner:

$$A(t,i) = \frac{1}{T} \sum_{n_t}^{n_t+T-1} a_i^2(n). \tag{2.12}$$

The binary mask that results in $S_{pyk}(t,i)$ uses only amplitude values that are selected from Eqs. (2.10) and (2.11).

$$S_{pyk}(t,i) = \begin{cases} A(t,i), & \text{if } F_w(t,i) \text{ satisifies (2.10)and(2.11).} \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

Figure 2.9 shows the binary mask and underlying gammatone amplitude estimates for a given speech sample.

The next step is to investigate overlap detection methods using Pyknograms. Discontinuities in the Pyknogram layout is an indication of interfering speech. An analogy for speech harmonic patterns are skiing tracks left behind on a snowy surface. In the single-speaker case, the patterns leave parallel tracks that progress relatively slowly over time and correspond to fundamental frequency harmonic tracks. In the presence of an interfering speaker, these patterns are distorted by similar but typically intersecting tracks, which adds sudden jumps along the time axis (as shown in Fig. 2.10 describing Pyknogram extraction). Since speakers are only capable of producing one fundamental frequency at each time instance, it is expected that the harmonic tracks should be consistent and point-wise parallel across time. This keeps harmonics parallel over short time intervals. The presence of a second speaker

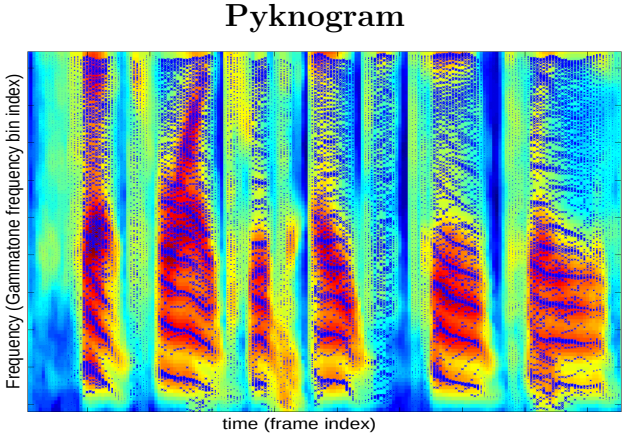


Figure 2.9. Pyknogram for a given speech signal. The spectrogram is plotted in the background for comparison. Pyknogram markers have been scaled by the amplitudes of corresponding $t$-$f$ units. Frequencies are scaled to equivalent rectangular bandwidth (ERB) rate. ©2017 IEEE

creates harmonic tracks that in general do not follow the same patterns, hence discontinuities are observed along time in Pyknograms. Therefore, Pyknogram variations across adjacent frames can be used to measure the presence of overlapped speech.

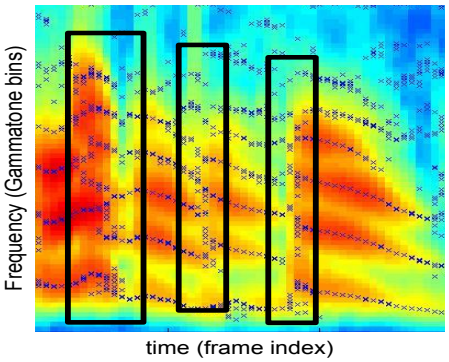


Figure 2.10. A closer look on Pyknograms for overlapped speech. The enclosed patches show discontinuities that occur in the presence of an interfering speaker. ©2017 IEEE

Figure 2.10 close-up of the Pyknogram for overlapped speech. The sudden jumps in Pyknogram frequency locations (shown in blocks) are an indication of two speakers.

### 2.4.3 Unsupervised overlap detection using Pyknograms

The average Euclidean distance between consecutive frames across all frequencies can be used to detect sudden jumps in Pyknograms along time. This has similarity to the technique used for spectral flux estimation (Rossignol and Pietquin, 2010). The distance function, $D_{ovl}$, at frame $t$ is computed as the 2-$norm$ distance between consecutive Pyknogram frames, $S_{pyk}(t,i)$ and $S_{pyk}(t-1,i)$.

$$D_{ovl}(t) = \sqrt{\sum_{i} \Big( \big( S_{pyk}(t,i) - S_{pyk}(t-1,i) \big)^2 \Big)}, \tag{2.14}$$

which results in a frame-based value for overlap distance. Later sections will investigate using longer time windows by averaging $D_{ovl}$ of adjacent frames.

Overlapped segments are expected to have higher $D_{ovl}$ values as compared to single-speaker speech. Figure 2.10 shows instances where sudden jumps are observed in the pyknogram of an overlapped signal. The average value of these distances for all frames in a speech segment corresponds to the amount of overlapped regions (higher values are associated with greater overlap).

The performance of the proposed detection metric will be evaluated on overlapped speech using the GRID database (Cooke and Lee, Cooke and Lee) (see Sect. 2.3 for more details on GRID). A key factor that determines the difficulty of detecting the presence of overlapped speech is the signal-to-interference (SIR) value. SIR is formally defined as the average energy of the foreground speaker to the energy of the background speaker, in dB. Of course, in the case of overlap detection, one speaker is not favored as the foreground over the other. Therefore absolute SIR values should be used for overlap detection. Greater absolute SIR corresponds to regions where one of the speakers has a lower impact on the signal energy. It is, therefore, more difficult to detect the occurrence of overlap in signals as the absolute SIR increases from zero. Henceforth, when we use SIR, we imply its absolute positive value.

Another important factor in detecting overlap is that the SIR value will change across sequential frames within a single file, a change which is due to the non-stationary nature of speech. This poses major restrictions on the effectiveness of overlap detection evaluation, since providing frame-based ground-truth becomes unrealistically difficult. One must therefore rely on ensemble measurements over complete speech files for which the average SIR is known. We therefore introduce the segment-based $D_{ovl}$ value, which is the ensemble average of all frames within $M$ second intervals.

$$^{M}D_{ovl}(t) = \sum_{t=t}^{t+\lfloor \frac{M}{T_s} \rfloor} D_{ovl}(t), \tag{2.15}$$

where $T_s$ is the length of the frame shift (in seconds), used here to determine the number of frames in an $M$ second interval. This notion is illustrated in Fig. 2.11, where $D_{ovl}$ distributions (histograms) extracted on a per-frame basis are compared with ensemble $D_{ovl}$ distributions associated with longer durations (2 seconds). $D_{ovl}$ for 2 second samples is calculated by averaging per-frame values. The "scores" ($D_{ovl}$ values) in Fig. 2.11 are pyknogram distances calculated using Eq. (2.14). The top figure (Fig. 2.11-a), shows the distribution of scores per *frame* (i.e., $25msec$ intervals) for overlapped (target) and clean (non-target/single-speaker) data. Figure 2.11-b shows the ensemble score distributions (average score over all frames within 2 second segments). The task in overlap detection is to separate the two classes in each plot (dark blue from light blue). As observed in these distributions, the per-frame classes are almost indistinguishable (Fig. 2.11-a), while in Fig. 2.11-b the classes show much better separation.

The observation in Fig. 2.11 is a characteristic of all non-stationary interferences, that frame-level detection is significantly less reliable compared to ensemble decisions. Therefore, longer durations should be used to detect the presence of overlap. Section 2.4.6 will investigate the relation between segment length (in other words, number of frames) and detection performance.

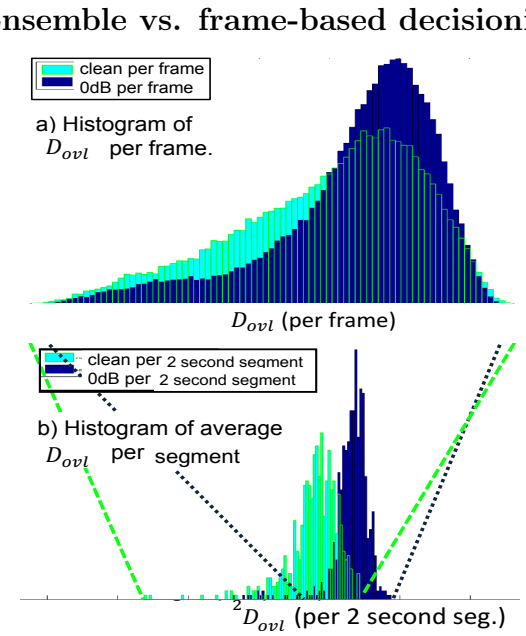


Figure 2.11. The effect of ensemble decisioning on distinguishability of overlapped regions. a) shows score per frame histograms and b) shows the histogram of ensemble scores. Using multiple frames to make a decision helps separate the distributions of clean and overlapped segments.

### 2.4.4 Evaluation

This section evaluates the proposed pyknogram-based overlap detection system in terms of *accuracy, robustness,* and *precision*. Evaluation tasks for each SIR category are in the form of standard binary classification problems, where target examples are from a collection of overlapped signals with fixed SIR values and non-target signals are clean (single-speaker). This section measures system performance using detection equal error-rates (EER; where false-positive and false-negative errors are equal). EER values are presented in Fig. 2.12 for different SIRs. The expectation is that the detection algorithm should be consistent across a range of SIR values (i.e., robustness). As for precision, we are interested to know how short signals can be before overlap detection performance significantly drops (noting the observation in Fig. 2.11).

### 2.4.5 Overlapped speech detection vs. SIR (Robustness & Accuracy)

Here the performance of pyknogram-based overlap detection is compared with the three baseline algorithms (described earlier in Sect. 2.2.1) across four SIR values. The goal is to monitor the changes in EER as SIR values increase. The experiments are designed to compare features in terms of how much separation they can create between single-speaker from overlapped speech. Overlapped signals are defined as target, while single-speaker signals (i.e., clean) are defined as non-target. The task is to perform binary classification using feature values as scores. The target/non-target signals used in this binary classification task are obtained from a pool of overlapped and single-speaker files. This task is repeated for each SIR condition separately to monitor the impact of SIR. In each task, overlapped signals with the same SIR are used as target examples and the overlap detection score (or feature value) assigned to them is compared against the scores estimated for clean files to

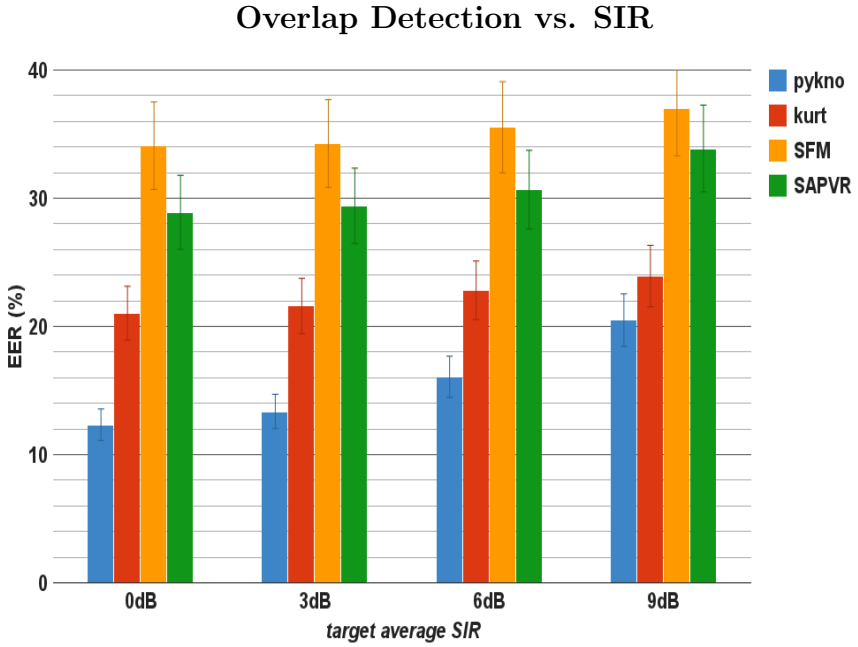


Figure 2.12. Overlap detection EER for different SIR values. The higher the SIR, the more difficult it is to detect the presence of interfering speakers. Shown here are results for four algorithms: pykno (Pyknogram-based – proposed method), kurt (kurtosis), SFM (spectral flatness measure), SAPVR (spectral autocorrelation peak-valley ratio).©2017 IEEE

compute the binary classification EER. Figure 2.12 compares performances for the proposed (pykno) and three baseline systems (kurt, SFM, SAPVR) across SIR values of $0, 3, 6$ and $9dB$. Pyknograms are shown to provide lower EER over the baseline features. This robust behavior across different SIR values is due to the resonance enhancement process, which takes place during Pyknogram extraction. Enhancing resonances reduces false alarm detections by removing the effects of non-harmonic components. These non-harmonic components are known to be confusing to overlap detection systems.

### 2.4.6 Overlapped speech detection vs. segment length

A main concern in dealing with overlapped regions is that overlap decisions are less reliable as segment lengths become shorter. This restricts algorithm precision in terms of the ability to detect overlap in a frame-based framework. It was shown in Sect. 2.4.3 that using multiple adjacent frames in the form of average $D_{ovl}$ (i.e., ${}^{M}D_{ovl}$) increases separability between single-speaker and overlapped speech distributions. The averaging defined in Eq. (2.15) can therefore be applied to our baseline features.

This method of treating non-stationary signals can help define the "precision" of proposed overlap detection algorithms. In other words, the question becomes: "what is the least number of frames required to maintain stable detection accuracy?"

Precision is most valuable in tasks such as speaker diarization in conversational speech, where overlap mostly occurs at speaker transitions between turn-takings. The goal of this analysis is to evaluate system precision and compare pyknogram-based detection with baseline features. It is useful to see how short can overlap segments get before observing a significant drop in system performance. Once again, overlap detection performance is measured through the detection EER. Figure 2.13 shows the change in system performance as shorter duration segments are used to obtain overlap decisions. It is shown that regardless of the feature used to detect overlaps, performance drops as fewer frames (shorter time segments) are used to make decisions. Furthermore, we see that performance stabilizes for all

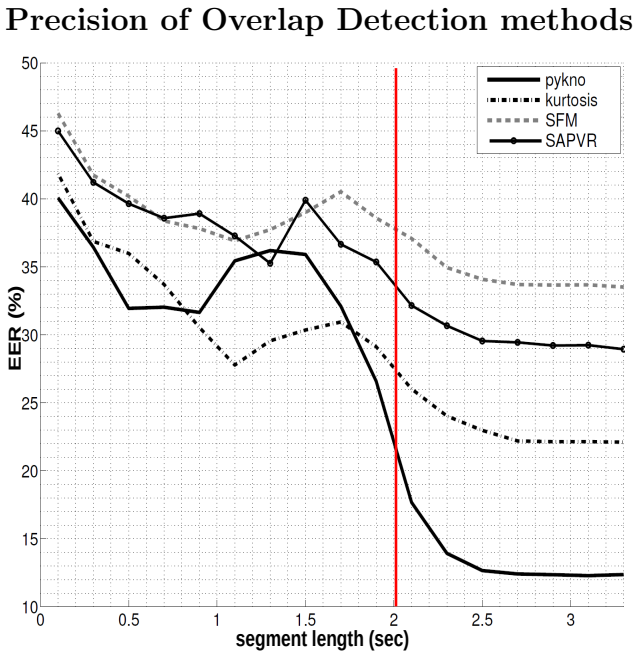


Figure 2.13. Overlap detection EER as a function of segment length. The plot shows that signal lengths should be at least 2 seconds for the algorithms to start reaching their best performance. ©2017 IEEE

features for segments 2 seconds and longer. While performance drops for all features for short segments, the proposed Pyknogram method maintains better performance compared to other methods as test duration decreases.

## 2.5 Gammatone Sub-band Frequency Modulation Spectra

So far, one of the two proposed overlap detection methods has been presented. The second method discussed in this chapter provides an alternative way of viewing overlapped speech. The theory and motivation behind the second proposed feature extraction are described in this section.

### 2.5.1 Motivation

This section begins with a brief analysis of frequency modulated (FM) sinusoids and their spectral characteristics. Modulating the frequency of a sinusoid results in a signal with

more frequency components than the original sinusoid. For example, the spectrum of a single-tone frequency modulated carrier contains frequency components that depend both on the amplitude and frequency of the modulating signal, both of which contribute to the modulation index, $\beta$ (Carlson and Crilly, 2010). Fig.2.14-a shows the spectrum of a single-tone FM signal. This signal is typically simplified and defined as,

$$x_c(t) = A_c cos(2\pi f_c t + (\beta sin(2\pi f_m t))), \tag{2.16}$$

where $f_m$ is the frequency of the modulating sinusoid and $A_c$ and $f_c$ are the carrier amplitude and frequency, respectively. In Fig.2.14-a, the amplitude of the $n^{th}$ frequency component of $x_c(t)$, considering $f_c$ as the origin, is the $n^{th}$ order Bessel coefficients at $\beta$ (i.e., $J_n(\beta)$).

Since overlapped speech consists of two speech signals, a closer analogy to the problem of overlapped speech is observed in the case where the modulating signal has more than one

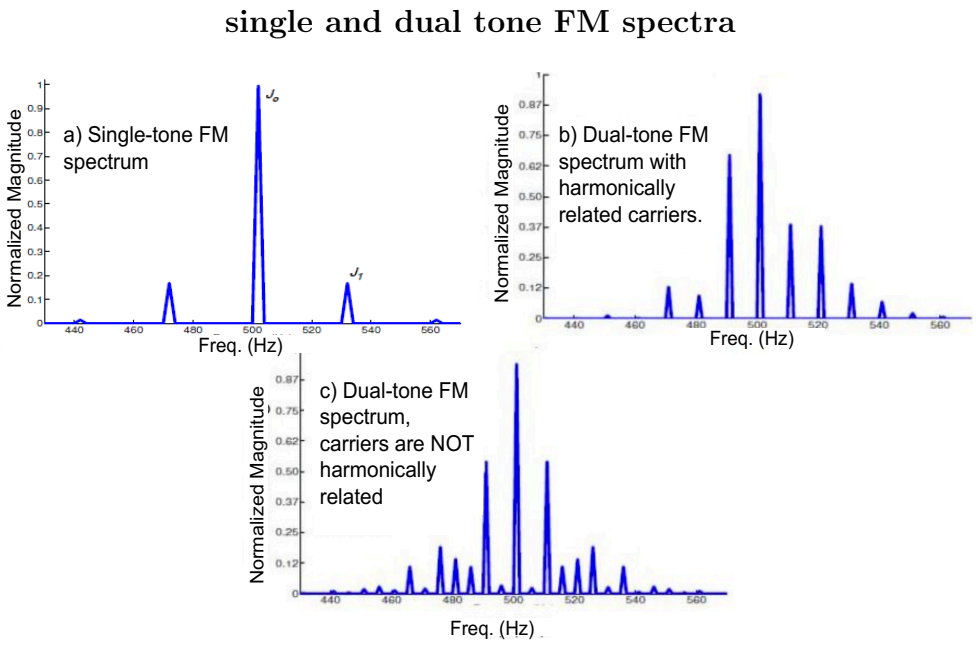


Figure 2.14. Comparing the dispersity of different FM signals. From top to bottom. a) Single-tone FM Spectral Magnitude $f_1 = 10$. b) Harmonically related double-tone FM Spectral Magnitude $f_1$=10, $f_2$=20. c) Not harmonically related double-tone FM Spectral Magnitude f1=10, f2=25

sinusoid (as seen in Eq. (2.17)). Fig.2.14-b and -c compare the spectra of double-tone FM signals in two scenarios. In Fig.2.14-b, the two tones are harmonically related (one is an integer multiple of the other), while in Fig.2.14-c, the frequencies of the modulating tones do not share a common integer factor. Consequently, the spectrum in Fig. 2.14-c is more disperse than that in Fig.2.14-b. The same conclusion can be made from Eq. (2.18) where the number of frequency components of the Fourier transform of $x_c(t)$ is greater when $f_1$ and $f_2$ are not harmonically related. An FM signal for a double-tone modulating signal with frequencies $f_1$ and $f_2$ can be represented as,

$$x_c(t) = A_c cos(2\pi f_c t + [\beta_1 sin(2\pi f_1 t) + \beta_2 sin(2\pi f_2 t)]). \tag{2.17}$$

Defining $J_n(.)$ as the nth order Bessel function, the Fourier series expansion of Eq. (2.17) can be compactly defined as follows,

$$x_c(t) = A_c \sum_n \sum_m J_n(\beta_1) J_m(\beta_2) cos(2\pi(f_c + nf_1 + mf_2)t). \tag{2.18}$$

This observation is particularly interesting, since in overlapped speech segments one of the major confusions is in distinguishing related harmonics (i.e., those that belong to the same speaker) versus non-related harmonics. Here, Eq. (2.18) suggests that the number of frequency components of $x_c(t)$ in a given range is greater when caused by non-harmonically related frequencies. On the other hand, if $f_1$ and $f_2$ are harmonically related, Eq. (2.18) can be simplified as,

$$x_c(t) = A_c \sum_{n'} K_{n'} cos(2\pi(f_c + n' f_1)t), \tag{2.19}$$

where $n'$ is defined such that for each $m$ and $n$ pair,

$$K_{n'} = J_n(\beta_1) J_m(\beta_2). \tag{2.20}$$

### 2.5.2 GSFM system description

Despite analyses on sinusoidal signals, a substantial difference between the spectral characteristics in multi-tone and single-tone FM signals versus speech is that in speech the spectrum consists of multiple harmonic components across its bandwidth. In order to interact with only a few sinusoidal components, a natural solution is to decompose the signals into multiple sub-bands by means of a filter-bank. As was the case for Pyknograms in Sect. 2.4.1, a gammatone filter-bank is used for sub-band analysis.[6] Sub-band outputs are used to modulate the instantaneous frequency of a sinusoidal carrier. Detecting overlapped speech with the use of multiple sub-bands results in multiple decisions for each speech segment. If a specific channel does not have the sufficient information to distinguish overlapped speech from single-speaker speech, the lack of information can be compensated for by other sub-bands. This framework results in a more consistent detection framework compared to a scenario where the system only relies on a single decision per frame. The GSFM feature extraction procedure is summarized in Fig. 2.15:

1. apply a gammatone fiterbank to the input speech signal

2. demodulate each sub-band to base-band (since the output for each channel is a band-pass signal and located around the center frequency of each sub-band). This is done by multiplying a sinusoid tuned at the center frequency of the gammatone sub-band followed by low-pass filtering.

3. Block the output signals over time into frames.

4. Compute the frequency modulated signal for each time-frequency unit by using the output of the previous step as the modulating signal.

[6]The gammatone filter-bank has been widely used in computational auditory scene analysis (CASA) literature to simulate the auditory periphery processing.

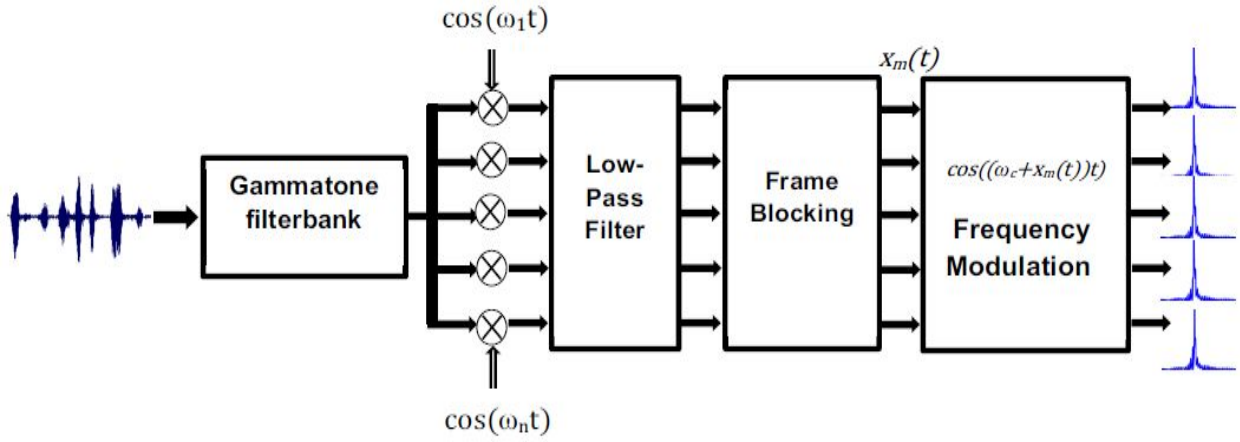


Figure 2.15. GSFM block diagram.

5. Use the spectral magnitude of the modulated signal as the output.

Given the output gammatone sub-band frequency modulated (GSFM) spectra, many operations could be used to quantify the amount of dispersion. A technique similar to that proposed in (Krishnamachari et al., 2000) is to use the relative peak amplitudes in the GSFM spectra. As shown in Fig. 2.14, the amplitudes of Bessel components drop more rapidly for harmonically related tones and even more so for single tones. The relative peak amplitude ratios can be used to represent spectral roll-off in the FM spectra for each time-frequency unit.

Figure 2.16 shows the difference in GSFM spectra for a typical time-frequency unit, $(t, i)$, in overlapped versus single-speaker speech. The GSFM roll-off is defined as:

$$R(t,i) = \sum_{k=2}^{N} \frac{P_1(t,i)}{P_k(t,i)} \tag{2.21}$$

where $t$ and $i$ respectively denote the frame and sub-band indexes. $P_k(t,i)$ corresponds to the $k^{th}$ peak in the GSFM spectrum with respect to the center peak (i.e., $P_1(t,i)$) that lies at the carrier frequency and is equal to the 0th Bessel coefficient (see Fig. 2.16). We call $R(t,i)$ the GSFM roll-off factor computed at each T-F unit, $(t,i)$. The collection of GSFM roll-off factors are used as a time-frequency representation for any given speech segment.

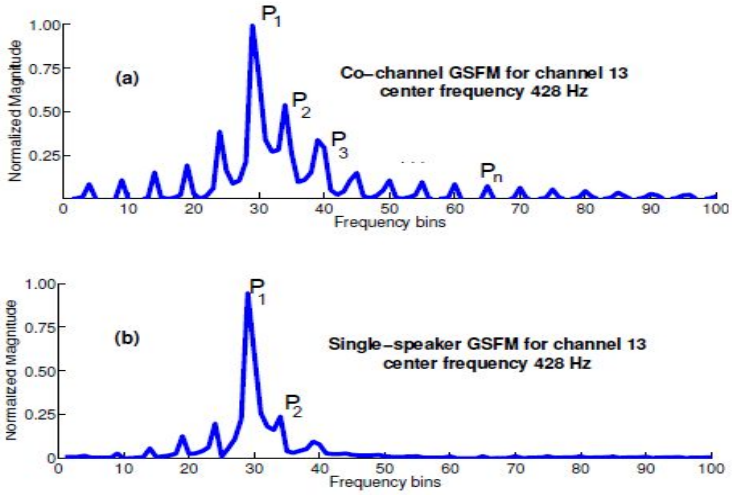


Figure 2.16. Comparison of GSFM spectra for overlapped speech and single-speaker speech. a) The GSFM of an overlapped speech segment at the 13th gammatone sub-band with center frequency 428 Hz. b) The GSFM of single-speaker speech at the same sub-band.

### 2.5.3 Unsupervised overlap detection using GSFM roll-off

The feature used to detect overlapped speech uses a combination of the information obtained from all sub-bands, since each band can provide a local decision as to whether a frame contains speech from more than one speaker. We can use the overall count of highly disperse time-frequency units (i.e., units with higher GSFM roll-off) per time unit as a decision score for the amount of overlapped speech in a given speech segment. Highly disperse roll-off values are obtained by applying a 2 class K-means clustering. K-means results in a threshold of the roll-off values, $R_{thr}$. By applying the threshold to the GSFM roll-off values (i.e. $R(t,i)$) for a given speech segment, one can estimate the decision score for overlapped speech detection. The decision score is calculated using the following equation:

$$S = \frac{1}{T} \sum_{\forall(t,i)} I\{R(t,i) > R_{thr}\}, \tag{2.22}$$

where

$$I\{x\} = \begin{cases} 1, & \text{if x is true.} \\ 0, & \text{otherwise.} \end{cases} \tag{2.23}$$

Here, T is the signal length in time units and $R_{thr}$ is the GSFM threshold obtained from clustering GSFM roll-off values into two sets, corresponding to both higher and lower values. The more the number of high roll-off values per unit time, the higher the likelihood of overlapped speech.

### 2.5.4 Evaluation

Overlap detection experiments for GSFM are conducted in a manner similar to what was presented in Sect. 2.4.4. Once again, it is useful to investigate features in terms of detection accuracy, robustness, and precision.

### 2.5.5 Overlapped speech detection vs. SIR (Robustness & Accuracy)

As explained before, a main challenge in overlap detection is addressing a large range of interference ratios, even within a single recording. Ideally, an overlapped speech detection system should perform consistently well in any SIR. However, this is neither feasible nor necessary. As noted earlier, the evaluation database used in this study consists of 4 different absolute SIR conditions. Figure 2.17 compares the performance of the proposed GSFM spectral roll-off feature with Pyknogram-based (pykno) and three baseline features (kurt, SFM, SAPVR). The figure shows how GSFM along with Pyknogram-based features outperforms the three baseline systems, while maintaining slightly lower error rates compared to the Pyknogram-based features.

### 2.5.6 Overlapped speech detection vs. segment length

All overlap detection algorithms presented in this study use the collective knowledge obtained from multiple adjacent time frames from a given speech segment to estimate a score that represents the likelihood of overlapped speech. The precision of the overlap detection system tells us how short a segment could be before we observe a substantial drop in performance. In Fig. 2.18, EER across different signal time lengths in 0dB average SIR. It was observed in Fig. 2.13 that the EER varies most in kurtosis for different signal durations. This is expected, since kurtosis is calculated from the $3^{rd}$ and $4^{th}$ order moments, which are less accurately estimated with insufficient data. From Fig. 2.18, we can also conclude that the breaking point for all systems is at approximately 2 seconds, which implies that overlap detection performance drops dramatically for segments less than 2 seconds long. The performance drop is more severe for GSFM features, since it uses the distribution of roll-off values in calculating the final score Eq. (2.22).


Figure 2.17. Overlap detection EER for different SIR values. The higher the SIR, the more difficult it is to detect the presence of interfering speakers.

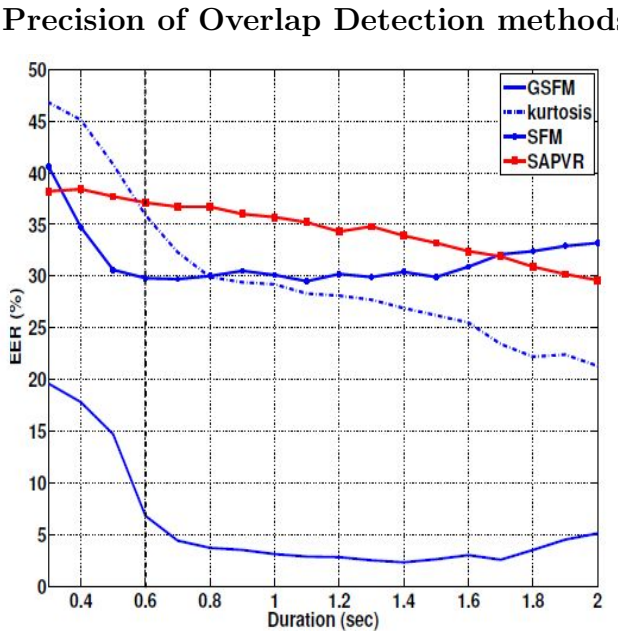


Figure 2.18. Overlap detection EER as a function of segment length.

## 2.6 Performance of Pyknogram vs. GSFM

Typical overlap detection systems are designed to detect overlapped speech segments in data that are generally clean of any environment background noise, which makes them less reliable for data collected in real meeting and conversation scenarios. This section investigates the scenario in which overlap detection takes place in noisy conditions. In other words, in addition to speaker interference, environment noise is also present in the data. For consistent performance in overlap detection, we use overlapped files with average signal-to-interference (SIR) of $0dB$, which means that the two utterances are mixed with the same average energy. So far, the SIR value has been our key assessment factor in overlapped speech detection performance (Sect. 2.4.5 and 2.5.5). Alternatively, this section keeps SIR constant and varies additive environment noise signal-to-noise ratio (SNR). To measure performance under noise, files are mixed with noise samples extracted from Prof-Life-Log (Ziaei et al., 2013) recordings with SNR values ranging from clean($100dB$) to $-10dB$. It is important to note in this context that the difference between SIR and SNR be clear to the reader. SNR

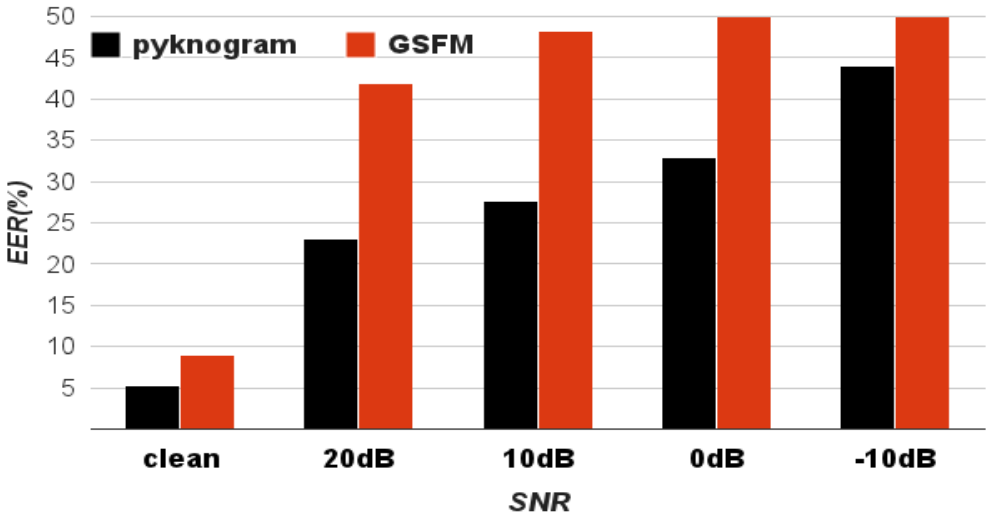


Figure 2.19. A comparison of overlap detection equal error rates (EER) for pyknogram (proposed) and GSFM-based systems for different amounts of added noise. It is clear that GSFM is vulnerable even to the slightest amount of noise (high SNR). ©2015 IEEE

specifies the amount of noise added to the files (overlapped or not) and SIR determines the relative energy of the two utterances in overlapped files. Figure 2.19 shows overlap detection EER values for different SNR values and compares the performance of Pyknogram-based features with GSFM features. As seen in the figure, GSFM performance drops dramatically even for the most trivial noisy condition ($20dB$). It is worth mentioning that in Sect. 2.5.5 it was shown that GSFM outperforms Pyknogram features as well as all baseline features.

Performances from other overlap detection algorithms were not included here, since none of the existing algorithms claim robustness in noisy conditions.

## 2.7 Summary

The focus of this chapter has been on overlapped speech, which refers to segments where both speakers (primary and interferer) are active. Independent cross-talk data (as defined in Chapter 1) were used in order to isolate overlap from other forms of interference, such as noise. Two overlap detection methods were proposed: Pyknogram, and GSFM. Pyknograms are useful for overlap detection, since they enhance speech harmonics regardless of the number of speakers in a signal. The significant difference between single-speaker and overlap harmonics allows Pyknograms to be an asset to overlap detection. The second method, GSFM, magnifies dispersive trends in speech sub-bands in the presence of overlap using multi-tone frequency modulation analysis. The two proposed methods are compared in real environmental noise and it is shown that despite outperforming Pyknograms in clean conditions, GSFM is highly affected by adding noise in overlap detection experiments. Therefore, Pyknograms provide more robust overlap detection performance, collectively from clean and noisy conditions.

# CHAPTER 3

# SPEAKER RECOGNITION IN OVERLAPPED SPEECH[1]

Detecting overlapped segments has previously been considered in speaker recognition diarization (Boakye, 2008; Yantorno, 1999). In such problems, the presence of a secondary speaker either decreases model reliability (in training), or introduces confusion in the decision-making process by distorting test files. Overlap detection is computationally advantageous when compared to enhancing the desired speaker's speech. This approach is especially useful when one has the luxury of neglecting overlapped data (Yantorno, 1999). Such is the case for speaker recognition and diarization (Boakye et al., 2008). By detecting overlapped speech segments, we are able to set them aside from the training and decision-making process (This assumes sufficient amount of data for train and test). However, overlap removal will not be the approach of choice in this chapter.

As pointed out in Chapter 2, the downside in removing overlapped segments is that a considerable amount of "usable" speech is also omitted from the speaker recognition system. An alternative solution is investigated in this chapter. In this solution, instead of directly applying overlap detection to data, overlap detection decisions are used as quality measure scores to assist speaker verification performance. This method is investigated in Sect. 3.4. The primary contribution of this study is therefore to:

- investigate overlap detection scores as quality measures for speaker verification.

First, however, this chapter begins by demonstrating the effects of overlapped data on speaker verification experiments (Sect. 3.1). This will provide an understanding of how overlaps affect speaker recognition. In addition, we will see how introducing overlaps to speaker verification affects train and test data separately (Sect. 3.2 and 3.3).

[1]Portions of this chapter were adopted from a journal article, soon to be published by IEEE, with the authors' full consent. Shokouhi, Navid, John H. L. Hansen, "Teager-Kaiser Energy Operators for overlapped speech detection," IEEE Transactions on Audio Speech and Language Processing ©2017 IEEE

## 3.1 Investigative setup

In order to show the detrimental effects of adding overlapped data to speaker verification, a case study is presented to analyze speaker recognition on data from the monaural speech separation challengev(SSC) described in Chapter 2 (Cooke et al., 2010). Experiments use 12-dimensional MFCC features (13 excluding the $0^{th}$ coefficient) plus $\Delta$ and $\Delta\Delta$, which together result in 36-dimensional features. The experiments use the Gaussian mixture model (GMM) approach where a trained speaker is modeled using a mixture of Gaussians and the maximum likelihood of a given test audio file is computed from the train model. GMM parameters are obtained through maximum a-posterior adaptation of the parameters of a speaker independent GMM (called a universal background model which is trained on a large pool of speakers). Only GMM means are MAP-adapted in these experiments. 512 mixtures were used to form the Universal background model (UBM).

The next section slightly digresses from overlapped speech and is intended for readers unfamiliar with speaker recognition frameworks. Readers that feel comfortable with the concepts of speaker verification, specifically the use of GMM-UBM setups for speaker for verification, may skip Sect. 3.1.1.

As mentioned above, trials are generated from train and test sets designed for the speech separation challenge. The amount of clean (i.e., single-speaker) training data for each speaker is approximately 15 minutes. Test data are partitioned into six SIR conditions, which are evaluated separately (see Sect. 3.2). The challenge also provides overlapped training data. Overlapped training data are used in Sect. 3.3 to train speaker models with the main speaker (i.e., model speaker in each train file) as the primary speaker and interfering speech from another randomly selected speaker. Experiments are gender-dependent, therefore the number of female speakers and male speakers is slightly different. In total, over 10000 trials are used to calculate equal error rates for each SIR condition presented in Sect. 3.2 and Sect. 3.3 with a target-to-imposter ratio of 0.001.

### 3.1.1 Speaker verification in a GMM-UBM setup

The speaker verification problem is a manifestation of the more generic problem of speaker recognition in the form of a binary detection problem. Speaker recognition, taken literally, implies sufficient knowledge of all speakers (or at least a large number of speakers) to "recognize" a given speaker from his/her voice. Speaker verification, on the other hand, tackles the more manageable problem of determining whether an audio segment was produced by a known speaker. A speaker verification system usually produces a likelihood value for each verification task (aka trial). The collection of trial likelihood values produces two distributions, one of which corresponds to true (aka target) trials and the other to imposter (aka non-target) trials.

The framework described above requires a model for the training speakers (for whom we have several recording sessions available), but for the test speakers we only use the features from a test audio signal to compute the maximum likelihood (ML) probability of the test speaker belonging to the corresponding train speaker in each trial. Figure 3.1 summarizes the speaker verification setup.

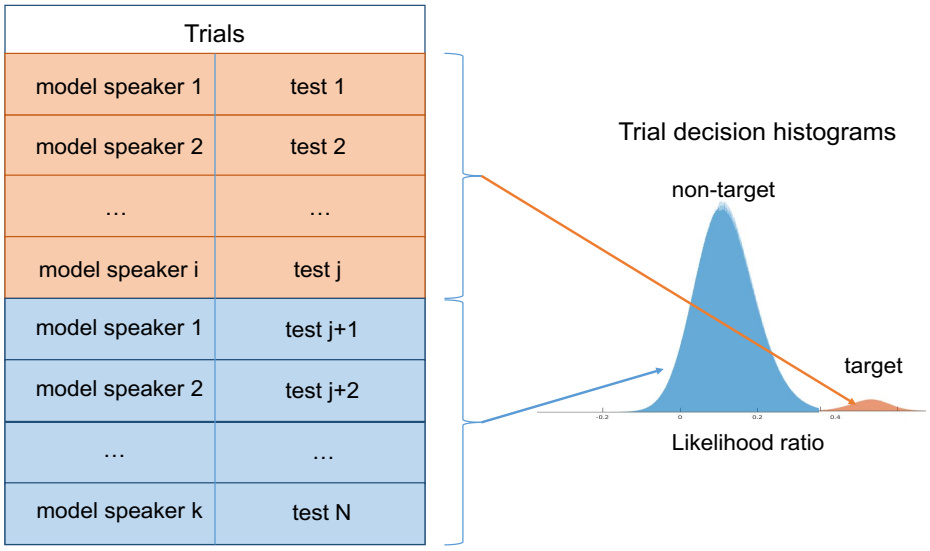


Figure 3.1. Speaker verification setup

The method through which likelihood ratios are calculated varies depending on the choice of model and system specifics. In the experiments presented in the next section, we use a GMM-UBM system (Reynolds et al., 2000). As mentioned earlier, the task of a verification system is to calculate the likelihood of a test audio belonging to a speaker model. In other words, if the test audio belongs to speaker $i'$ and the model (i.e., training speaker) belongs to speaker $i$, the objective is to calculate the likelihood of $i = i'$. Here, a model is a Gaussian mixture model (GMM) obtained by adapting the means of a universal GMM (aka UBM). The UBM is trained on a separate set of background development data. In the examples provided in Sect. 3.2 and 3.3, the development data is TIMIT. The verification system considers two hypotheses: 1) the probability of the test audio being generated from the given speaker model (i.e., $GMM_i$), and 2) the probability of the test audio being generated from the UBM, which in this case represents every other speaker except $i$. The ratio between these two ratios is used to quantify the system decision for a given trial. Figure 3.2 illustrates the procedure to determine the likelihood ratio for one trial.

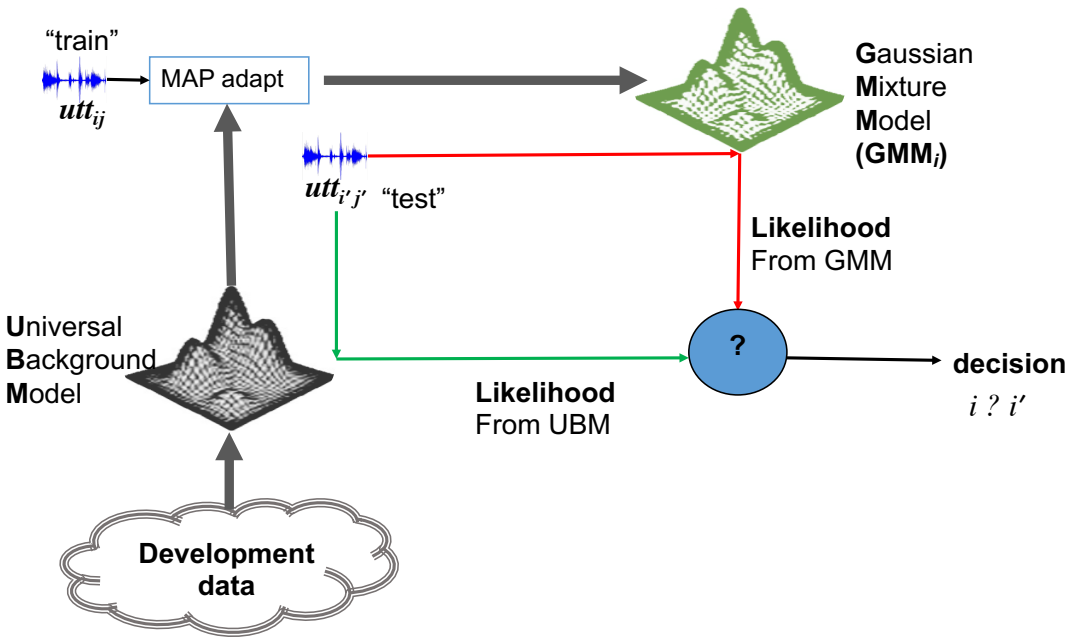


Figure 3.2. Speaker verification setup

Given this introduction, we expect that the reader should now be able to follow the remainder of this chapter. Useful descriptions of current topics on speaker verification and recognition can be found in (Hansen and Hasan, 2015).

## 3.2 Overlaps in test data

As a comparison benchmark, performance is evaluated under clean train and test conditions on the speech separation challenge (SSC) data. Gaussian mixture models (GMM) are adapted from a Universal back model (UBM) trained on TIMIT files (Sadjadi et al., 2013). For each model (training) speaker, there are 500 utterances in SSC, which are all used in the training process. Test files are available in all SIR conditions. As is expected, lower SIR values correspond to higher equal error rates. The presence of a secondary speaker, clearly causes confusion in the score distribution, leading to less separability between target and imposter trials. Recognition performance under clean test files and those with average SIR ranging in $+6, +3, 0, -3, -6, -9dB$ are provided in Fig. 3.3.

It is worth mentioning that it was tempting to compare these results with stationary noise experiments. However, contrary to expectations, it was observed that performances were better in the overlapped condition when compared to white Gaussian noise and speech-shaped noise interference, even for negative SIR values. This is most likely due to a misunderstanding caused by comparing stationary and non-stationary noise through the same measurement procedure, which is the SIR (or SNR). For a given target speech file, adding a certain amount of stationary noise will affect all frames, whereas in the case of non-stationary noise (here speech) only a portion of the frames over time receive non-uniform interference. This leads to incomparable results under presumably similar conditions which we decided to exclude from this study to avoid confusion. Therefore, an important take-away message here is that a certain **SIR** value does not translate to the same **SNR**.

### 3.3 Overlap in train data

This section examines the effect of adding overlapped speech to train files (see Fig. 3.4). Figures 3.5 and 3.6 compare the effects of adding overlapped speech in train and test files.

An interesting observation is the higher rate with which the EER increases when the SIR drops for the test condition. We believe this is due to the fact that in train conditions, the training of Gaussian mixture models tends to cancel out the effect of the secondary/interfering speech. For each speaker, the GMM is trained on a set of features, some of which are influenced by the desired speaker and the rest influenced by the interfering speakers. Since multiple training files are used to model each speaker (different training files have different interfering speakers), the GMM tends to converge to a common locale in the feature space, which ideally will belong to the speaker for whom the models are being trained. We call this effect "averaging out" (or cancelling out) of the interfering speakers. This, to some extent, slows the growth in EER as the data becomes noisier in train files. Such cancellation, however, does not exist across test files.

The results provided in Sect. 3.2 and 3.3 were provided to focus explicitly on the impact of overlaps on speaker verification. This is believed to be novel from two perspectives: 1) The data used here is actually overlapped, meaning that files contain two speakers speaking at the same time, and not the generic case of co-channel. Therefore, the results provide greater motivation and direction as we move on to subsequent sections, which focus on overlap detection. 2) The comparison made between adding overlap to test vs. training data provides insight in determining which (train or test) to prioritize to remove overlap. It should be noted that simply removing everything that is assumed to be overlapped does not necessarily yield optimal performance, since losing train/test data lowers model reliability and the number of test features used to calculate output likelihood values.

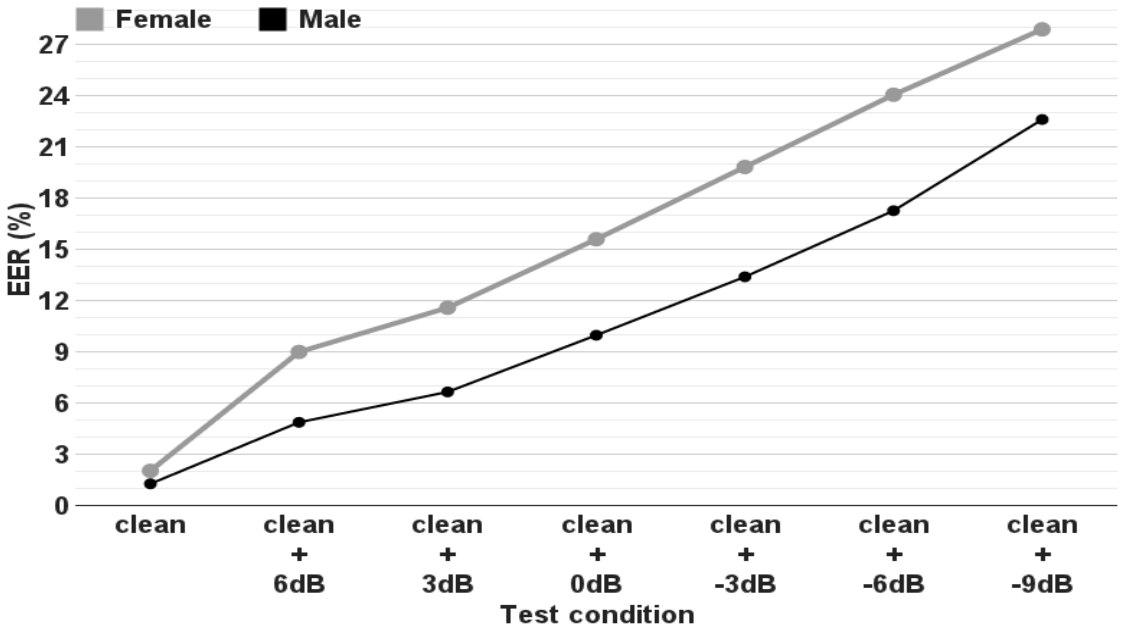


Figure 3.3. The rise in EER values as we increase the effect of overlapped speech (via decreasing the SIR). Starting from clean (i.e. single-speaker speech) to lower SIR values. The graph shows the case where train files are clean, but test files contain overlaps. ©2017 IEEE

### 3.4 Overlap detection as meta-data for speaker recognition

Using meta-data to yield more accurate decisions is a common practice in speaker verification evaluations (Brümmer and de Villiers, 2013; Mandasari et al., 2013). Incorporating quality measures such as speech activity detection (SAD) and effective file durations can significantly improve verification performance (Mandasari et al., 2013; Hasan et al., 2013) regardless of system architecture (be it i-Vector, GMM-UBM, or any other system). A quality measure here refers to additional information (i.e. meta-data) used alongside speaker verification likelihood ratios (aka speaker verification scores) to quantify the test and train conditions in trials. Meta-data provides lower-level scores that help increase distinguishability between target/imposter trials. For speaker verification in overlapped speech, the primary source of confusion is caused by the presence of interfering speakers. This section proposes to

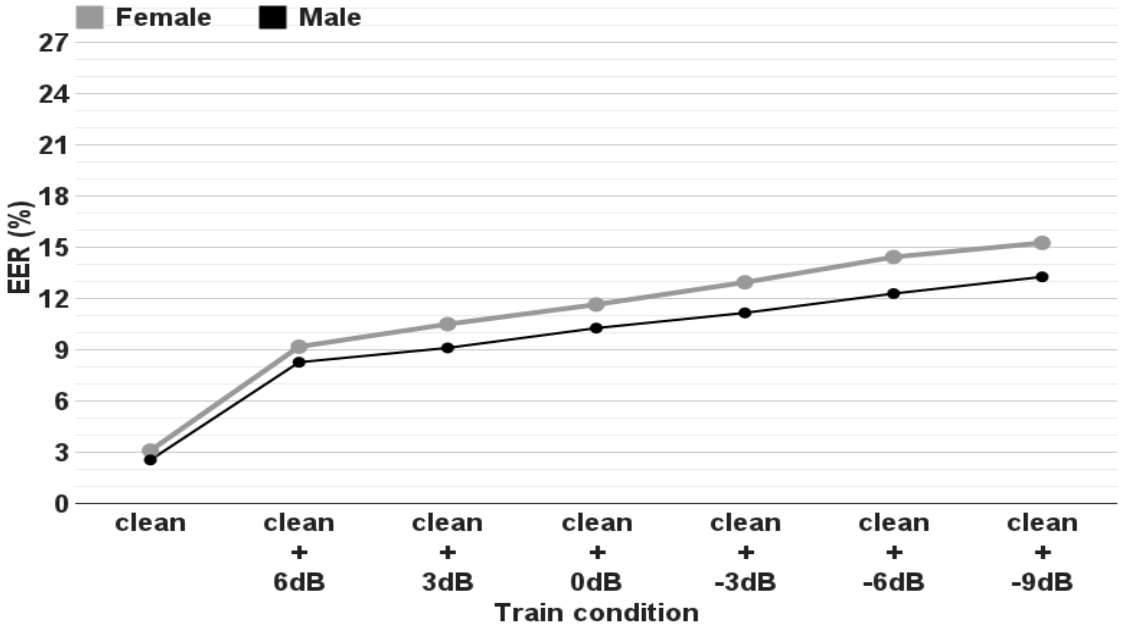


Figure 3.4. Female and male speaker verification experiments with clean (100dB SIR) test files, but train files contain overlaps. ©2017 IEEE

use scores from overlap detection algorithm(s) as secondary information to improve overall speaker verification performance.

There are several approaches through which quality measures can be applied in a binary classification scenario (Brümmer and de Villiers, 2013; Kryszczuk and Drygajlo, 2009; Kelly et al., 2013). Here, we use a stacking approach, called Q-stack, in which the quality measures (here overlap decisions) are concatenated (i.e., "stacked") along with speaker verification decisions (Kryszczuk and Drygajlo, 2009). The resulting vector is a high-dimensional score vector which allows more separability due to the additional information provided by the stacked dimensions. The stacked score vectors are then processed with a support vector machine (SVM) classifier. SVM parameters are trained using a development set extracted from a separate subset of the data. In experiments, the development set consists of $10,000+$ trials, a quarter of which are clean trials and the remaining $7,500+$ trials contain overlapped test files with either $0, 3, 6dB$ SIR levels. The target-to-imposter ratio in speaker verification

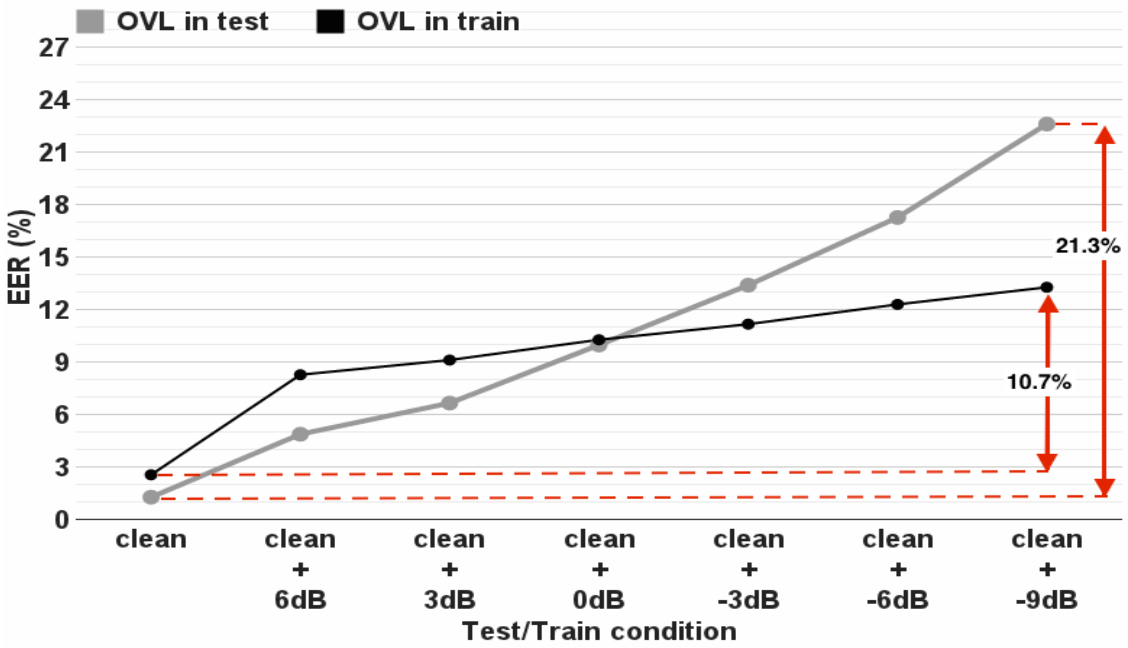


Figure 3.5. Comparing the impact of increasing overlap (OVL) in train vs. test data by decreasing SIR values. Experiments for male speakers. Lower SIR drops the performance more rapidly when applied to test data. ©2017 IEEE

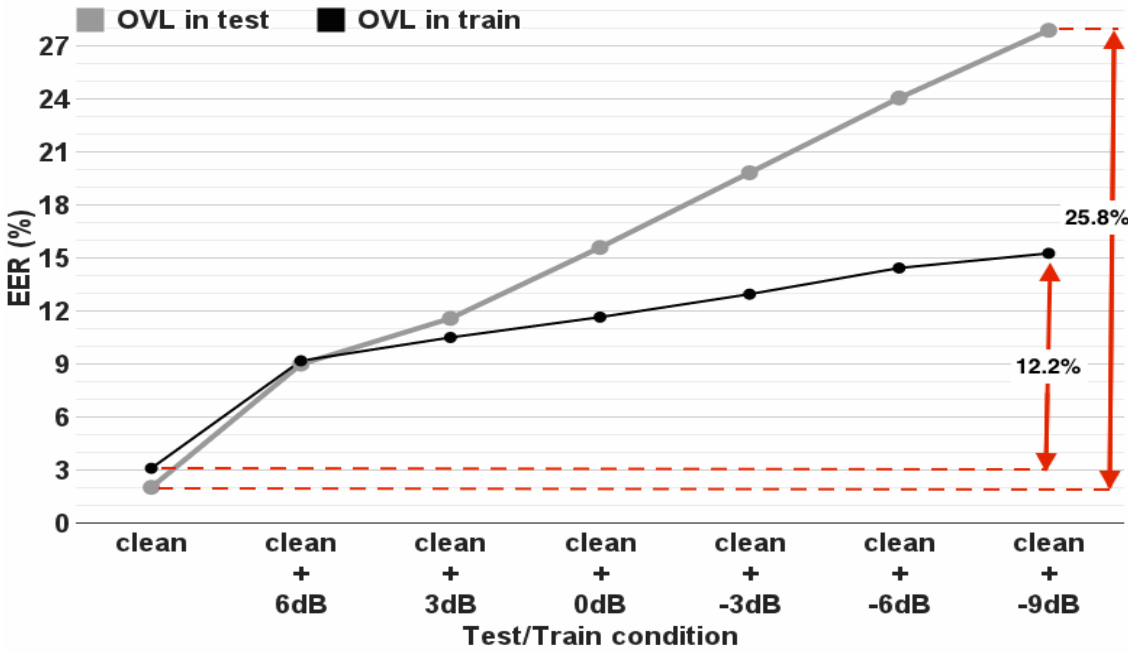


Figure 3.6. Shows the counterpart experiments of Fig. 3.5 for female speakers. ©2017 IEEE

Table 3.1. Speaker verification performance (EER) with and without overlap detection scores as meta-data. Grey cells highlight the features used in each experiment. The relative change in EER is presented in the last column. ©2017 IEEE

| raw GMM/UBM scores | pykno | kurtosis | SFM | SAPVR | ***EER (%)*** |
|---|---|---|---|---|---|
| ✓ | | | | | 11.36 |
| ✓ | ✓ | | | | 10.19 |
| ✓ | | ✓ | | | 13.51 |
| ✓ | | | ✓ | | 28.35 |
| ✓ | | | | ✓ | 9.48 |
| ✓ | ✓ | ✓ | | | 10.20 |
| ✓ | ✓ | | ✓ | | 10.47 |
| ✓ | ✓ | | | ✓ | 9.57 |
| ✓ | ✓ | ✓ | ✓ | | 10.31 |
| ✓ | ✓ | ✓ | | ✓ | 9.18 |
| ✓ | ✓ | ✓ | ✓ | ✓ | **9.10** |

trials is 0.001. An evaluation set of size $18,000$ trials with similar specifics and target-to-imposter ratio is used to test overall system performance.

Table 3.1 shows the improvements obtained by using the overlap detection scores individually and combined. Kurtosis and SFM show less correlation, however they provide significant complementary information when combined and used alongside SAPVR and pyknogram features. The best result is obtained when all four features are concatenated, since each overlap detection system may yield better performance in certain scenarios.

The better individual performances seen from SAPVR is because it is superior in distinguishing harmonic structures. Since speaker identities are mostly influenced by voiced speech, this assists the speaker recognition task in quantifying the amount of reliable voiced segments. Pyknogram-based detection is designed to locate harmonic discontinuities as opposed to the presence of harmonics.

Experiments show that the best performance is obtained using an SVM with a radial basis function (RBF) kernel. The SVM parameter(s) (here $\gamma$) are determined through cross-validation on the development data set. Class weights (i.e., target/imposter weights for the SVM classifier) and the cost (aka slack) parameter are selected according to the detection cost

function (DCF) parameters ($C_{fa}$, $C_{miss}$, and $prior$) used throughout experiments (Brümmer and de Villiers, 2013).

An additional experiment is also conducted using ideal overlap labels (labels from ground-truth) in the Q-stack paradigm which results in a lower bound for the performance. The EER for this lower bound is 8.74% (23% relative improvement). In the Q-stack algorithm, the relative drop in EER from using all overlap features is approximately 20%, which is not far off from when ground-truth labels are used. This confirms the effectiveness of the selected overlap detection features/scores. Using the proposed Pyknogram-based overlap detection system and baseline features achieves a relative improvement of 20%, a mere 3% lower than the best achievable performance provided by the oracle lower bound.

## 3.5 Summary

Chapter 3 has provided an investigative study on overlapped speech in speaker recognition. This effect was measured by adding overlapped speech to training and test data in a speaker recognition problem. Since the focus in this chapter was on physical attributes of overlapped speech, all experiments were conducted on independent cross-talk data (as defined in Chapter 1). It was shown that overlapped speech is more detrimental when added to test data, due to its impact on the final verification score. When overlap is added to training data, the effect of interfering speech is mitigated as the different secondary speakers are averaged out across training sessions provided for a given speaker. A proposed method to decrease the impact of overlapped speech was to use overlap detection scores as meta-data for a speaker verification system. The traditional approach has been to simply remove overlapped segments. Alternatively, this study has investigated the improvement obtained by preserving data (be it overlapped or not) while calibrating speaker verification scores to adjust to various levels of overlap in both test and training data.

# CHAPTER 4

# SPEAKER RECOGNITION IN CO-CHANNEL SPEECH

This chapter will address the problem of speaker recognition for co-channel recordings, rather than speaker recognition in overlap. The main focus is on interference from secondary speakers, be it overlapped or not. Secondary speech interference, the main characteristic of co-channel, is an important source of error for all automatic speech processing systems. Speaker recognition experiments are highly influenced by the presence of secondary speakers, due to reduced reliability of the trained models. Although the target speaker is a common factor in all training samples for a given speaker model, the standard structure of speaker recognition systems has not been designed to effectively average out interfering speech. With this in mind, the contributions of this chapter are to:

- systematically dissect the effects of overlap versus co-channel speech on speaker verification,
- investigate three proposed methods to improve probabilistic linear discriminant analysis for speaker recognition for co-channel speech.

As far as this study is concerned, few if any studies address speaker recognition in co-channel speech signals. However, the effects of artificially adding **overlapped speech** in a speaker verification setup has received considerable attention (Yantorno, 1999; Yantorno et al., 2000; Shao and Wang, 2003), as shown in Chapter 3. In many of these studies, the approach has been to automatically detect and remove overlapped segments from co-channel speech in training and test data. Although many overlap detection algorithms have been investigated over the years (Boakye et al., 2008; Shokouhi et al., 2013; Smolenski and Ramachandran, 2011; Krishnamachari et al., 2000), none have considered solving the problem in the more general case of co-channel interference.

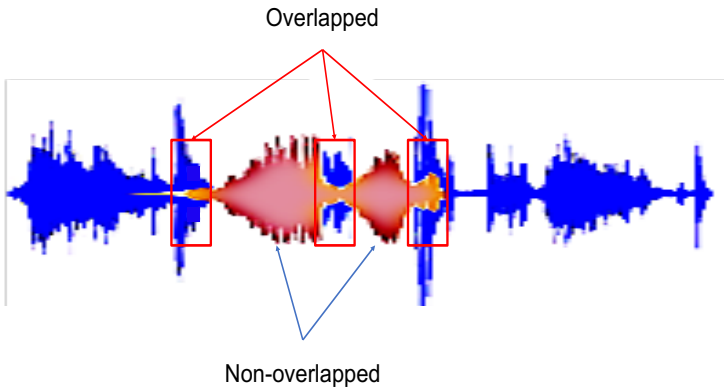


Figure 4.1. Difference between co-channel and overlapped speech. Overlap refers to instances where more than one speaker is active. Co-channel is defined as an entire stream that contains multiple speakers. Blue is the primary speaker. The figure shows that in a single-channel audio, the amount of non-overlapping co-channel data is typically more significant in conversational speech.

This chapter differentiates co-channel speech from overlapped speech by considering the latter to be a special case of co-channel where both speakers are active at the same time. Co-channel speech refers to the broader case where speakers are not necessarily overlapping (see Fig. 4.1). Here, the focus is on speaker recognition in co-channel speech interference, in which overlaps may occur. This sheds light on a more realistic problem, since only a small percentage of conversational speech contains amounts of overlap that are large enough to significantly impact speaker recognition performance (Cetin and Shriberg, 2006; Smolenski and Ramachandran, 2011). An example is given in Fig. 4.1, where the overlaps as well as non-overlapping speech interference are labeled. It can be seen that speech from the secondary speaker in a single-channel conversation recording is more likely to appear as non-overlapping segments.

A commonsense solution to co-channel interference for speaker recognition would be to separate speech from unwanted speakers within the original signal. The difference between what we present here and existing studies in speaker recognition for co-channel speech is that we would like to bypass solutions that require removing interfering speech from the original signal. Such solutions are primarily known as speaker diarization. Speaker diarization is

defined as the task of determining "who spoke when?" within an audio recording. Using diarization as a preprocessing step for speaker recognition in co-channel speech involves determining speaker identities in short segments within each recording, a task which is itself a speaker recognition problem. Alternatively, we are interested in modifying the model parameters extracted from co-channel data in a way that would only represent the primary speaker.

The solution we would like to focus on in this chapter is a model-based approach, rather than signal-based (e.g. speaker diarization). Currently, the most common form of modeling speaker-dependent features are i-Vectors (Dehak et al., 2011). I-Vectors are latent parameters that model the covariance of speaker/session-dependent Gaussian mixture models (GMM) with respect to a generic GMM (aka Universal background model – UBM). The UBM is ideally both session- and speaker-independent. The use of i-Vectors, has become a standard way of modeling speaker specific traits for speaker recognition. In many cases i-Vector extraction is considered a preprocessing step in performing speaker recognition.[1] Therefore, it is both reasonable and desirable to concentrate on post i-Vector analysis to deal with co-channel speech interference (Greenberg et al., 2012). The goal of this study is to build upon the latent variable perspective, popularized by i-Vectors (Dehak et al., 2011) and its predecessors (Kenny, 2010), to improve speaker recognition in co-channel signals. Working in the latent variable subspace provides the luxury of short-circuiting speaker diarization, a computationally intensive and potentially error-prone solution.

To further refine our problem statement, the following ground rules are set. It is assumed that: 1) Sufficient data is available from multiple recording sessions to train speakers; 2) Co-channel data is used in the evaluation set. In the standard i-Vector speaker recognition framework, often a number of recordings are provided for each speaker. These i-Vectors can then be projected onto a subspace using probabilistic linear discriminant analysis (PLDA) (Prince

[1]See Appendix

and Elder, 2007) to compensate for channel variations across different recordings (Kenny, 2010; Garcia-Romero and Espy-Wilson, 2011).[2] Therefore, latent variables in the PLDA subspace are calculated in a way to only represent speaker-dependent information (Matejka et al., 2011; Cumani et al., 2013; Burget et al., 2011; Lei et al., 2012). Now if i-Vectors are to be extracted from co-channel signals, the speaker-dependent latent variables from PLDA must represent a combination of all speakers in the original audio file. For the case of speaker recognition in co-channel speech, the task of our proposed system would be to also account for the fact that i-Vectors might have been extracted from co-channel sessions.

This chapter investigates using modified versions of the PLDA paradigm to make i-Vectors collected from co-channel sessions suitable for speaker recognition experiments. The goal is to create overall robustness with respect to interfering speech. PLDA uses inter- and intra-session variabilities from a development set to find a subspace in the i-Vector space that best represents speaker dependencies. Here we investigate the possibility of performing an i-Vector normalization strategy by considering co-channel interference to be a form of inter-session variability. It is important to us that our experiments be easy to replicate and require minimal additional information (labels, speaker and channel information, etc.).

An investigative approach to the effects of co-channel speech in speaker recognition is presented in the next section, Sect. 4.1. This section shows how much performance could drop when co-channel data is added to speaker verification experiments. In addition, Sect. 4.1 also compares the impact of overlap and co-channel on speaker verification. This comparison is made to show the importance of addressing co-channel, which is more general, rather than overlap for a large group of speaker recognition problems. In Sect. 4.2, standard PLDA and its following modified version, simplified PLDA, are described. It is described how channel compensation is performed through these methods (Prince and Elder, 2007; Kenny, 2010).

[2]Channel variation refers to differences in recording conditions and devices. Readers are reminded not to confuse channel information with co-channel speech.

The two-covariance interpretation of PLDA (Ioffe, 2006) is also presented in Sect. 4.2, this interpretation motivates the proposed *co-channel aware PLDA model*. Section 4.3 investigates treating co-channel interference in a manner similar to how PLDA addresses channel mismatch, using a background data preparation scheme we call *mixed PLDA*. Section 4.4 proposes another modification to PLDA, called *dual-eigenvoice PLDA*. Dual-eigenvoice PLDA (or dePLDA) removes redundancy in the eigenchannel matrix of PLDA by replacing it with a second eigenvoice matrix. Section 4.5 proposes *co-channel aware PLDA*, which is used to remove speaker interference from PLDA's speaker-dependent latent variable subspace.

## 4.1 Effect of Co-channel in Speaker Verification

The first step in addressing co-channel speech in speaker verification is to establish how much performance degradation is expected. A number of studies have investigated "co-channel speech" and its effect on speaker verification. However, each provides different insight due to the somewhat nuanced definition of co-channel, as explained in the introduction. Many consider overlap synonymous to co-channel, an equivalence which is strongly argued against in this study. A clear distinction has been made between overlap and co-channel, a distinction that makes addressing co-channel speech in speaker verification more important than overlap in some regards.

As we will see in this section, overlap contributes to a small portion of the total errors compared to co-channel in many large-scale speaker recognition problems. Furthermore, overlap detection and removal is an error-prone approach and many have pointed out that for the purpose speaker recognition, a strict removal of overlaps is not necessarily an ideal solution (Smolenski and Ramachandran, 2011). For example, (Yantorno, 1999) shows that co-channel speech in the form of overlaps significantly increases equal-error-rates for speaker verification systems based on Gaussian mixture models (GMM). An interesting result presented in (Yantorno, 1999) shows that keeping all "usable speech" rather than removing all

overlaps yields better performance under co-channel. Therefore, usable speech detection was proposed instead of overlap detection to improve speaker verification performance (Shao and Wang, 2003; Wu et al., 2003). In a sense, usable speech refers to speech from the foreground speaker (speaker of interest) with high signal-to-interference ratio and/or all voiced segments of the foreground speaker in which spectral harmonic patterns have not been severely disrupted (Smolenski and Ramachandran, 2011). Another analytic study on overlap in speaker verification was presented in (Shokouhi and Hansen, 2016) by comparing the impact of overlap in the test data with overlap in training data. The authors argue that an averaging effect occurs when multiple instances of overlapped training data is provided in enrollment sessions, while test data usually has a more direct role in deriving likelihood ratios for each trial. An alternative to removing overlapped segments for speaker recognition has been to perform speaker separation (Saeidi et al., 2010; Mowlaee et al., 2010), or in some cases simultaneous identification of both speakers in an overlapped stream (Zhao et al., 2015; Sadjadi and Heck, 2014). Many of these studies focus on overlapped speech rather than the more general case of co-channel speech.

Although overlap presents an undeniably difficult challenge in speaker verification, the amount of overlap in conversational co-channel speech is far too small to significantly impact speaker verification in large-scale problems. Later in this section, we will separately evaluate system performance under overlap-only conditions. First, it would be useful to determine exactly what percentage of everyday conversational data contains overlaps. Readers are encouraged to visit (Shriberg et al., 2001) for a detailed analysis of overlaps in conversational speech corpora.

To investigate the amount of overlap in conversational speech, two popular corpora are examined here:

- Switchboard2: a large collection of $\approx 5$ minute telephone conversations involving several hundred speaker from across the United States.

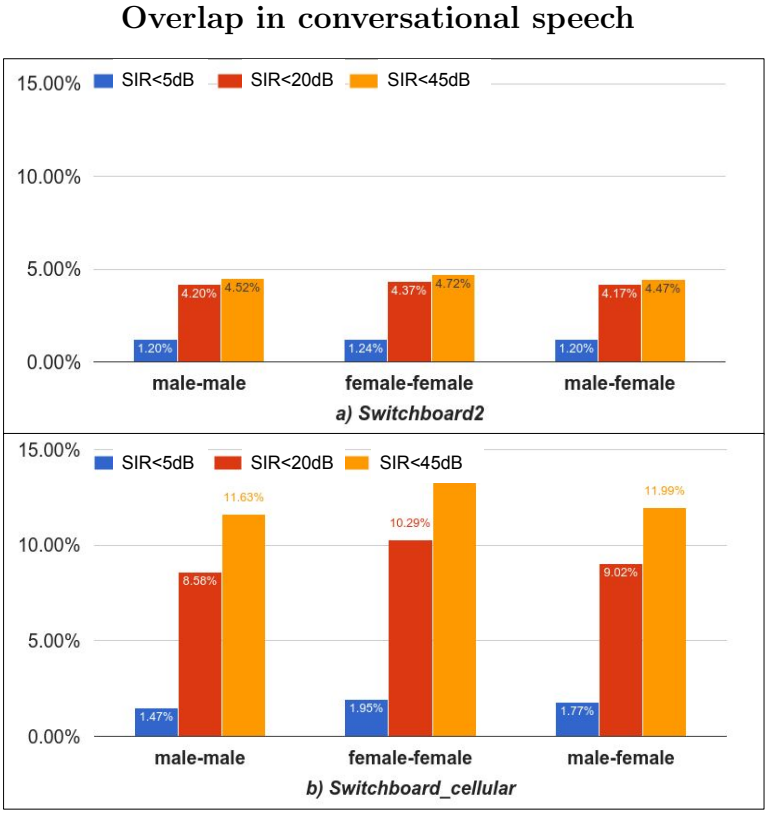


Figure 4.2. Percentage of overlaps to total speech in Switchboard2 and Switchboard cellular telephone conversations. Three SIR upper bounds are selected to label overlaps; 5dB, 20dB, 45dB. The higher the SIR upper bound, the stricter the overlap labels. Separate results are shown for male-male, male-female, and female-female conversations.

- Switchboard Cellular: a collection of $\approx$ 5 minute telephone conversations on cellular phones.

- AMI meeting corpus: A dataset consisting of 100 hours of meeting recordings from several locations across Europe.

For each session, the separate (almost interference-free) channels provided for each speaker are first segmented into speech and silence using an energy-based speech activity detection. Signal energies are required for each time-frame (25$msec$ window), since signal-to-interference ratio (SIR) is used to define overlap.

$$SIR(n) = 10log_{10}(\frac{P_1(n)}{P_2(n)}), \tag{4.1}$$

where $P_1(n)$ is the per-frame energy of channel 1 (i.e., the primary speaker) and $P_2(n)$ corresponds to channel 2 (the secondary speaker). The variable $n$ represents frame indexes. Channels are mixed (per sample addition of the two signals) to create co-channel data. Instances at which both speakers are active are considered overlapped. Since the SIR value varies for different segments, we use a threshold on the SIR to label frames as overlapped. Segments with an absolute SIR (i.e., $|SIR|$) lower than the threshold are considered overlapped. The amount of overlap varies with the maximum allowable SIR (i.e., threshold) set by the evaluator. For example, one might consider the mere presence of two speakers at the same time sufficient to label a segment as overlap, which is an indication of high SIR thresholds ($45dB$ in Fig. 4.2). A more pragmatic view, however, is to choose an SIR small enough to preserve as much data as possible. This way of preserving some overlapped segments is shared in many studies under the definition of usable speech (Yantorno, 1999; Smolenski

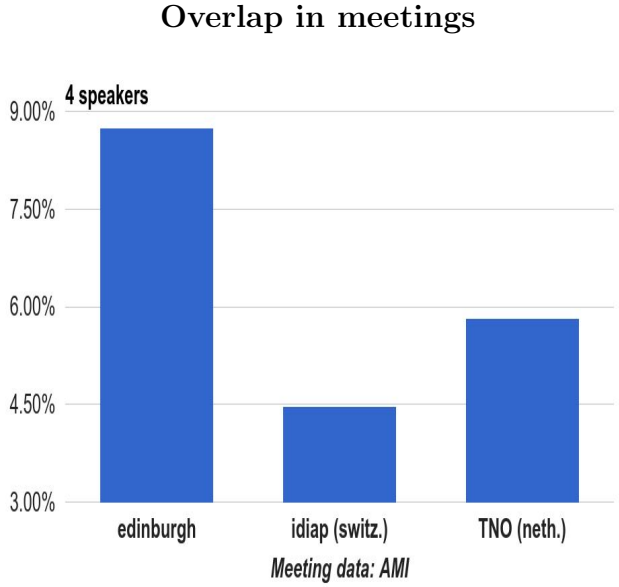


Figure 4.3. Percentage of overlaps to total speech in the AMI meeting corpus. All meetings used here have exactly 4 speakers. The percentage of overlap is significantly higher here compared to Switchboard (compare with blue bars in Fig. 4.2).

and Ramachandran, 2011). Lower SIR values, $(0-5)dB$, have a more significant impact on speaker verification. However, to provide more insight, three SIR upper bounds have been used in Fig. 4.2; 5, 20, $45dB$. Any overlap up to $5dB$ should have noticeable impact on speaker verification. An upper bound of $45dB$ is also chosen, since the percentage of overlap is constant beyond $45dB$. It is shown in Fig. 4.2 that with a $5dB$ threshold, the percentage of overlap to total speech is below 2%. For the curious reader, the overlap percentages for three groups of conversations are provided: male-male, male-female, and female-female pairs. It is clear that for Switchboard gender does not play a role in dictating overlap percentage.

For a different perspective, the AMI meeting corpus is also analyzed. The difference between meetings and phone conversations is in the number of speakers and face-to-face interaction. The speculation is that the number of speakers increases overlap, while on the other hand the fact that all speakers are present in the same room (i.e., face-to-face interaction) limits the amount of overlap. Another difference is that SIR is not as well defined for meetings as it is for two-party phone-calls, since multiple parties may be active at the same time. Figure 4.3 assumes that at least two speakers should have a relative SIR of up to $5dB$. Any additional speaker is evaluated with respect to the primary speaker, but with a $20dB$ threshold. As Fig. 4.3 suggests, location also plays a significant role in overlap percentage. It is better to refrain from speculating the impact of location, since such analyses exceed the scope of this study.

In text-independent speaker recognition, where we are interested in long-term acoustic characteristics, 4-5% of overlapped speech in our data has little effect on speaker recognition accuracy. The point here is not to say that overlapped speech can be neglected in speaker recognition, but to clarify that speaker recognition in its most common form is more concerned with co-channel speech in general rather than *overlap* as it appears in everyday English conversations. In the more general case of co-channel speech interference, the presence of secondary speakers has a significant impact on speaker verification. We can roughly

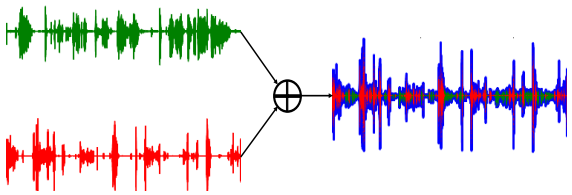

Figure 4.4. Mixing two channels of a Switchboard phonecall. The example here mixes the signals with 0dB SIR. Blue shows the resulting co-channel signal. Red and green each show one of the single-speaker signals.

estimate that in a two-party phone conversation, approximately 50% of the data contains the unwanted secondary speaker (compare this with 2% in Fig. 4.2). Section 4.1.1 will investigate the effect of overlap and co-channel on speaker verification EER.

### 4.1.1 Co-channel Interference in Trials

In this section, a series of speaker verification experiments are conducted on Switchboard2 to demonstrate the effect of using co-channel speech in enrollment and test data. These experiments are not introduced as "baseline", since it is an unfair assessment to add co-channel interference to trials and expect PLDA to perform well. The purpose of this section is to show the increase in equal error rates as a result of co-channel speech. We also further emphasize the point made in Sect. 4.1 by identifying the impact of overlap from co-channel interference on the EER.

For these experiments, single-speaker data is used to estimate PLDA parameters (not co-channel data). Trials are evaluated at different levels of co-channel interference (i.e., SIR level). In each scenario, trial recordings are summed with their counterpart channel from the phone conversation to create co-channel data, as if speakers are speaking on a single channel (shown in Fig. 4.4). Speaker labels for trial recordings are generated based on the primary speaker. In this context, foreground speaker refers to the speaker of interest. For example, in a $5dB$ co-channel session generated from Switchboard2 containing speakers $\mathbf{X}$ and $\mathbf{Y}$, if $\mathbf{X}$ were the foreground speaker, the average energy of $\mathbf{X}$ would be $5dB$ higher

than the average energy of $\mathbf{Y}$. Five SIR levels are chosen throughout experiments; $100dB$ (i.e., clean sessions), $20dB$, $10dB$, $5dB$, and $0dB$. In $0dB$ the average energy of the primary and secondary speakers is equal. To avoid mismatch, the clean condition is also generated through the same procedure with an SIR of $100dB$ favoring the primary speaker.

A gender-independent universal background model (UBM) is created using 8kHz single-speaker NIST SRE data from 2004, 2005, and 2006 challenges (NIST, 2004, 2005, 2006). The UBM consists of 2048 Gaussian mixtures representing a 39 dimensional feature space (13 dimensional MFCC plus $\Delta$ and $\Delta\Delta$). The same data from SRE 2004-6 is used to estimate a total variability (TV) matrix, which extracts 400 dimensional i-Vectors (Dehak et al., 2011). The data used here to estimate PLDA parameters are single-speaker recordings from NIST SRE 2008 (NIST, 2008). PLDA training data consists of approximately $11k$ single-speaker utterances from over 1300 speakers. Trial data is developed from 2500 Switchboard2 recording sessions containing approximately 800 speakers. Prior to feature extraction, trials are processed using ComboSAD, an unsupervised speech activity detection (Sadjadi and Hansen, 2013). ComboSAD has previously shown to provide stable performance improvement in such speaker recognition tasks (Hasan et al., 2013).

Figure 4.5 shows speaker verification performance for the five SIR cases. As shown, EER for the clean condition (i.e., $100dB$) is significantly lower than all the other SIR levels, even $20dB$. The sudden jump in EER shows the significance of co-channel interference.

A second experiment is conducted to separate performance drop caused by overlap. To show this, all speech from the secondary speaker is removed from the recordings, except for segments that overlap with the foreground speaker. This is accomplished by using voice activity detection (VAD) labels from the 100dB trials, while using $0dB$ audio data for the trials. Figure 4.6, compares speaker verification under overlap with 0dB co-channel speech. The figure shows that overlap plays a small part in the rise of EER (red bar) compared to co-channel interference (yellow bar).

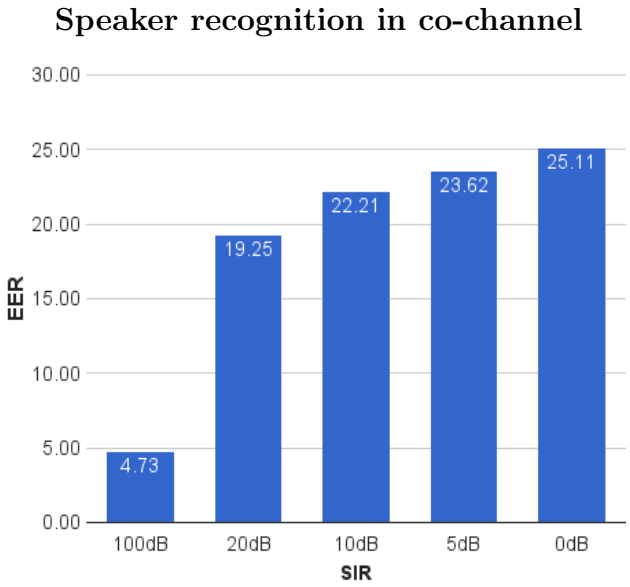


Figure 4.5. Speaker verification performance with co-channel speech in switchboard trials. The i-Vector/PLDA system uses a typical system configuration and is fully trained on single-speaker data. The purpose of this chart is to show the rapid increase in equal error rate (EER) as co-channel data is added to the trials. 100dB SIR represents clean (single-speaker) trials.

## 4.2 Motivation

Before presenting the proposed methods, it is essential to provide a brief overview of the chronological introduction and development of probabilistic linear discriminant analysis (PLDA) in speaker recognition. PLDA was initially proposed for face recognition in (Prince and Elder, 2007). It was later adopted as a channel compensation step for speaker recognition using i-Vectors (Kenny, 2010).[3] A number of studies since then have presented different formulations for the factor loading paradigm most commonly known as PLDA. In this study, we are specifically interested in three formulations: 1) standard PLDA (Kenny, 2010), 2) simplified PLDA (Matejka et al., 2011; Garcia-Romero and Espy-Wilson, 2011), and 3) the two-covariance model (Brümmer and De Villiers, 2010). The nomenclature used here

[3]I-Vectors (Dehak et al., 2011) are fixed-length vectors generated from audio recordings and are used to model speaker-dependent information of said recordings. Although i-Vectors can be directly used to compare speakers (for example using cosine distance), removing channel-dependent information is an important step in improving performance for speaker recognition using i-Vectors. See Appendix for more information on i-Vector based speaker recognition.

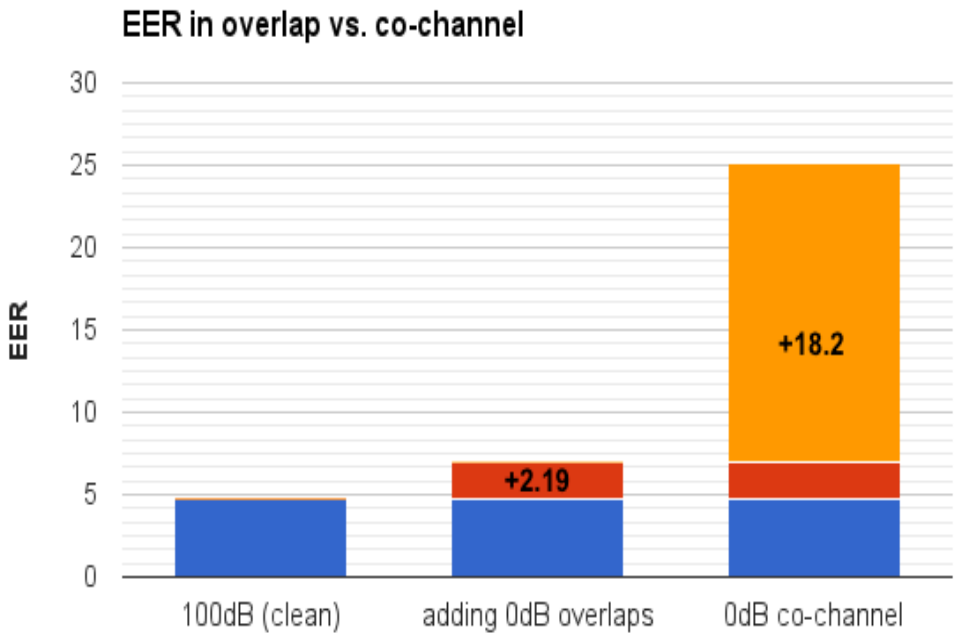


Figure 4.6. Comparing the effect of overlap in speaker verification with the more general case of co-channel. This study differentiates overlap from co-channel speech by considering overlaps to be segments during which both speakers are active. Co-channel refers to the more general case of two speakers in an audio stream, not necessarily overlapped (see Fig. 4.1). The chart shows that overlap plays a small part in the rise of EER compared to co-channel interference.

is adopted from a recent study by Sizov et al. (Sizov et al., 2014) aimed at unifying the variations proposed for PLDA over the past decade.

### 4.2.1 Standard PLDA

The general idea of probabilistic linear discriminant analysis is to find a subspace in the i-Vector space that best represents speaker-specific components. The search for this subspace is based on a training dataset organized in a way that emphasizes differences between speakers as well as variations of each speaker across different recordings (aka sessions). The data organization comprises $n_i$ observation i-Vectors for speaker $i$ from a set of development speakers. PLDA assumes the following linear factorization for each i-Vector $\mathbf{m}_{ij}$:

$$\mathbf{m}_{ij} = \mathbf{m}_g + \mathbf{V}\mathbf{y}_i + \mathbf{U}\mathbf{x}_{ij} + \mathbf{z}_{ij}, \qquad j = 1, ..., n_i \tag{4.2}$$

where $\mathbf{m}_g$ represents the global i-Vector mean. Speaker- and session-dependent latent variables, $\mathbf{y}_i$ and $\mathbf{x}_{ij}$, take a standard normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$. $\mathbf{V}$ and $\mathbf{U}$ are typically tall matrices representing eigenvoice and eigenchannel subspaces, respectively. Eigenvoice refers to the collection of factor loadings (represented in $\mathbf{V}$) that construct the speaker-dependent subspace. Eigenchannel refers to the session-dependent subspace. In addition to the eigenchannel subspace a session-dependent and normally distributed slack variable, $\mathbf{z}_{ij}$, is included to express session variabilities. In Eq. (4.2), $\mathbf{z}_{ij}$ takes a diagonal covariance matrix, $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma_d})$, (Kenny, 2010; Prince and Elder, 2007). PLDA predicts model parameters, $(\mathbf{V}, \mathbf{U}, \mathbf{\Sigma_d})$, using the expectation-maximization (EM) algorithm (Prince and Elder, 2007). After estimating subscpace components using background development data, trial i-Vectors are reduced to the same speaker-dependent subspace using PLDA and scored through a hypothesis testing procedure (see (Prince and Elder, 2007) for details). The hypothesis testing stage estimates the likelihood ratio that states whether two trial i-Vectors (train and test) belong to the same speaker, or if they belong to two different speakers.

### 4.2.2 Simplified PLDA

The second formulation reduces complexity in Eq. (4.2) using the fact that session-dependent latent variables ($\mathbf{x}_{ij}$ in Eq. (4.2)) are not directly used in the scoring process. Therefore, as long as models are able to effectively estimate the eigenvoice subspace, channel-dependent components (essentially all non-speaker dimensionality) are redundant. With this in mind, the second term in Eq. (4.2) is removed in simplified PLDA and all channel information is captured in the slack variable. The slack variable in this case is assumed to have a full covariance matrix.

$$\mathbf{m}_{ij} = \mathbf{m}_g + \mathbf{V}\mathbf{y}_i + \mathbf{z}_{ij}^f. \qquad j = 1, ..., n_i \tag{4.3}$$

The use of a full covariance matrix in the slack variable can be interpreted as combining the diagonal slack covariance in Eq. (4.2) with the eigenchannel subspace projection $\mathbf{U}\mathbf{U}^T$ (Sizov et al., 2014). In this study, the simplified PLDA formulation is considered the baseline formulation.

### 4.2.3 PLDA as an extension to LDA

The last interpretation, called the two-covariance model (Sizov et al., 2014), is a probabilistic extension of linear discriminant analysis (LDA). The two-covariance model describes the i-Vector space in terms of between- and within-speaker covariances, as does LDA. It is well known that LDA models feature spaces as a mixture of Gaussians, in which each mixture has the same covariance, $\mathbf{\Phi_w}$. Gaussian mixtures represent within-class (i.e., session) variability, therefore $\mathbf{\Phi_w}$ is referred to as the within-class covariance matrix. LDA is commonly used to find the optimal discriminating subspace of a given set of training speakers, relative to their within-speaker variation (Ioffe, 2006). The problem, however, is that the aforementioned subspace is only optimal for the given training speakers. What LDA fails to provide is a continuous (or in this context, stochastic) representation of each mixture's centroid.[4] Therefore, centroids are considered deterministic in LDA. When a new speaker is introduced, the resulting projection of LDA is not necessarily reliable. PLDA provides a stochastic representation of class centroids using a between-class covariance matrix, $\mathbf{\Phi_b}$ (Ioffe, 2006). The centroid distribution assumes a continuous centroid subspace, which acknowledges the possibility of unseen speakers. PLDA can therefore be defined as a combination of two distributions;

[4]In this model, centroids in the i-Vector space are known to correspond to speakers. In other words, each centroid represents a speaker. The assumption here is that if enough session variability is provided for a set of i-Vectors that belong to the same speaker, the mean of i-Vectors in that set is an accurate representation for the speaker.

1. the distribution of i-Vectors in each class representing a certain speaker, which is a Gaussian with mean $m_c$, mean of a speaker's i-Vectors, and covariance $\mathbf{\Phi_w}$. The subscript $c$ here represents the class label, or speaker identity. Therefore, the probability distribution of any given i-Vector, $\mathbf{m}$, assuming that it comes from class $c$ is:

$$\mathbf{m} \sim \mathcal{N}(\mathbf{m}_c, \mathbf{\Phi_w}|c), \tag{4.4}$$

2. the class centroid distribution is also assumed Gaussian:

$$\mathbf{m}_c \sim \mathcal{N}(\mathbf{m}_g, \mathbf{\Phi_b}), \tag{4.5}$$

   where $\mathbf{m}_g$ is the global mean of all class centroids (assumed to be equal to the global mean of all i-Vectors) and $\mathbf{\Phi_b}$ is the between-speaker covariance matrix.

Defining channel variability as a function of speaker variation helps PLDA model unseen speakers (i.e., speakers that are not present in the development set). As opposed to LDA, which is incapable of offering optimal solutions to speakers that are absent from the training set (Ioffe, 2006). The interpretation in this section provides a perspective which will be used in the proposed method (Sect. 4.5) to investigate adding co-channel interference as a contributor to within-class variability. Using a transformation matrix ($\mathbf{V}$), equations Eq. (4.4) and Eq. (4.5) can be translated to Eq. (4.3) (Sizov et al., 2014) in which the within- and between-covariances are diagonalized (Ioffe, 2006).

Now that the bases for understanding PLDA were introduced, the next sections will introduce the three proposed methods of this study. The first, mixed PLDA, is not a significant departure from PLDA and only replaces the choice of training data to adapt to co-channel speech. The second method, dual-eigenvoice PLDA, is based on the standard and simplified PLDA models to remove redundancies while modeling secondary speakers in co-channel data. The final method, co-channel aware PLDA, uses the two-covariance interpretation to improve the modeling of co-channel data.

## 4.3 Proposed method: mixed PLDA

The search for eigenvoice and eigenchannel subspaces involves a careful selection of development data. The idea in data preparation for PLDA is to provide sufficient channel diversity for each speaker to model within-speaker variations, while maintaining high speaker counts to model between-speaker variability. Channel and speaker variation introduced in the development data are directly translated into within- and between-speaker covariances, respectively. These covariances are used to estimate PLDA parameters, Eq. (4.2) and Eq. (4.3) (Sizov et al., 2014). The data-driven perspective towards channel compensation using PLDA has inspired a number of studies to address other types of variability through the same data selection procedure, where instead of channel diversity one could generate a development set with age (Kelly et al., 2013; Kelly and Hansen, 2016) or language diversity (Misra and Hansen, 2014) (in the case of multi-lingual speakers). Therefore, our first approach is to investigate co-channel speaker recognition performance when background PLDA data contains co-channel speech recordings.

Co-channel recordings for each speaker are obtained from separate conversations. In these conversations, it is important to maintain diversity in secondary (aka interfering) speakers; since without sufficient secondary speaker diversity, the PLDA model will train to both primary and secondary speakers. Figure 4.7 is a diagram of how the PLDA data is arranged in this approach, called *mixed PLDA*. Mixed PLDA introduces speaker interference in the background development data using co-channel mixtures to implicitly train the PLDA model by recognizing speaker interference as session variability.

Mixed PLDA describes the use of co-channel background data in simplified PDLA, Eq. (4.3). This will be used as a method to provide fair comparison with our proposed PLDA formulation described in the next section.

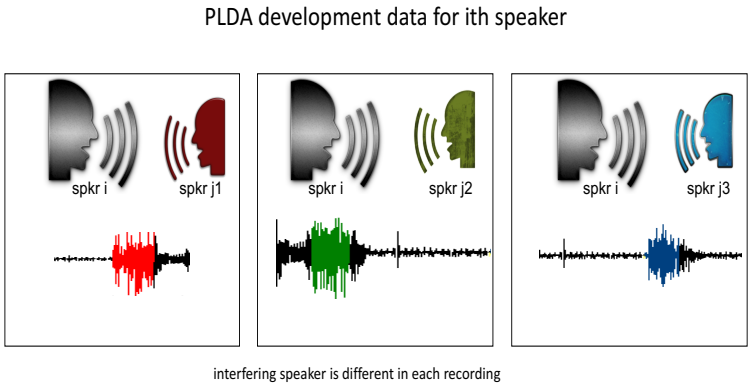


Figure 4.7. Creating development data for co-channel aware PLDA. the mixed PLDA approach uses co-channel data for each speaker in the background model. Recordings for the $i^{th}$ speaker consists of co-channel sessions with different speakers.

### 4.3.1 Co-channel Interference in Trials with mixedPLDA

Now that *mixed PLDA* has been introduced, its performance is evaluated by adding co-channel data to PLDA training sets. There are a number of ways co-channel data can be inserted, particularly in the choice of SIR. Three cases are investigated in the experiments:

- $0dB$: All co-channel data used to train PLDA is $0dB$.
- $(0, 100)dB$: Half of the data is clean and the other half is $0dB$.
- $(0, 5, 10, 20, 100)$: co-channel files are uniformly selected from one of these five SIR values.

Table 4.1 shows that mixed PLDA with half $0dB$ and half $100dB$ co-channel training data for PLDA causes the least damage in the clean condition (a.k.a $100dB$), from 4.73 to 5.22. Mixed PLDA (0,100$dB$) also reduces error rates the most. Minimum detection cost functions (minDCF) is calculated over the operating point ($C_{fa} = 10, C_{miss} = 1, prior = 0.001$). These observation are consistent with those reported in (Shokouhi and Hansen, 2015).

Table 4.1. Mixed PLDA: PLDA performance when co-channel interference is introduced as session variation, without changing the original PLDA formulation. The EER for simplified PLDA is presented in the last column for comparison.

| SIR (dB) | mixed PLDA (0dB) | | mixed PLDA (0,5,10,20,100dB) | | mixed PLDA (0,100dB) | | simplified PLDA | |
|---|---|---|---|---|---|---|---|---|
| | EER (%) | minDCF | EER (%) | minDCF | EER (%) | minDCF | EER(%) | minDCF |
| 100 | 7.48 | 0.587 | 5.71 | 0.809 | **5.22** | 0.472 | 4.73 | 0.407 |
| 20 | 19.75 | 0.897 | 17.49 | 0.865 | **17.14** | 0.872 | 19.25 | 0.909 |
| 10 | 21.79 | 0.941 | 20.24 | 0.923 | **19.82** | 0.918 | 22.21 | 0.944 |
| 5 | 23.20 | 0.960 | 21.86 | 0.948 | **21.02** | 0.940 | 23.62 | 0.967 |
| 0 | 24.61 | 0.976 | 23.13 | 0.971 | **22.43** | 0.963 | 25.11 | 0.985 |

It is interesting that the best performance is obtained with a 50-50 split of co-channel and clean data in the PLDA training set. At this point, we have no clear explanation as to why *mixed PLDA* $(0, 100dB)$ is superior to $(0dB)$ or $(0, 5, 10, 20, 100dB)$.

## 4.4 Proposed method: dual eigenvoice PLDA

The approach in *mixed PLDA* relies on PLDA's channel compensation capabilities. PLDA recognizes the variabilities observed across different recordings for a given speaker and removes them from the speaker subspace (aka the eigenvoice factors in $\mathbf{V}$). Therefore, *mixed PLDA* treats interfering speech as channel mismatch. This is accomplished by adding co-channel i-Vectors to the PLDA background data and leaving it to the PLDA model estimation process to recognize interfering speech and remove its effects in the speaker subspace. However, interfering speech is not effectively picked up by neither the eigenchannel matrix (in case of the standard PLDA in Eq. (4.2)) nor the slack variable, in Eq. (4.3). We address this by introducing our second approach which is to add a speaker dependent term intended to model interfering speech. Since the second term also corresponds to speaker identities, it should as well be represented by the speaker subspace. We call this approach *dual eigenvoice PLDA*, for which the PLDA factorization is:

$$\mathbf{m}_{ij} = \mathbf{V}\mathbf{y}_i + \mathbf{V}'\mathbf{w}_{ij} + \mathbf{z}_{ij}. \tag{4.6}$$

Since the model still needs to represent channel variabilities, we keep $\mathbf{z}_{ij}$ as a full covariance normal distributed vector. $\mathbf{V}'$ represents the speaker dependent subspace, as does $\mathbf{V}$, with the difference that one is a rotation of the other with an unknown rotation factor. $\mathbf{w}_{ij}$ is the latent variable which corresponds to the interfering speaker. There are a few reasons that prevent us from directly using $\mathbf{V}$ as factor loadings for the interfering components in Eq. (4.6), one being that as part of the degrees of freedom in the PLDA solution, the eigenvoice matrix can only be determined up to an unknown rotation matrix (Sizov et al., 2014) (similarly for $\mathbf{V}'$). Equation Eq. (4.6) uses different notation for the second eigenvoice matrix to remind us of this limitation. This prevents directly using $\mathbf{V}$ to represent the interfering speaker term. However, since we know that $\mathbf{V}$ and $\mathbf{V}'$ must be related by a rotation matrix, we can use this knowledge to simultaneously update $\mathbf{V}$ and $\mathbf{V}'$ in the EM iterations.

As discussed above, the relation between $\mathbf{V}$ and $\mathbf{V}'$ is characterized by an unknown rotation matrix, $\mathbf{R}$:

$$\mathbf{V} = \mathbf{R}\mathbf{V}'. \tag{4.7}$$

A reasonable $\mathbf{R}$ can be estimated via singular value decomposition (SVD) by considering the columns of $\mathbf{V}$ and $\mathbf{V}'$ as data points in the speaker-dependent subspace. The rotation matrix is derived from the cross-variance between the basis functions of $\mathbf{V}$ and $\mathbf{V}'$, $\mathbf{S}$:

$$\mathbf{S} = \sum_{i=1}^{N_V} \tilde{v_i} \tilde{v_i'}^T, \tag{4.8}$$

where $\tilde{v_i}$ is $v_i$ are the eigenvoice basis vectors centered at the origin and $N_V$ is the number of columns in $\mathbf{V}$ (and/or $\mathbf{V}'$, since both have the same dimensions). The rotation matrix is defined as below:

$$\mathbf{R} = \mathbf{S}_{row}\mathbf{S}_{col}^T, \tag{4.9}$$

where $S_{col}$ and $S_{row}$ are the column and row spaces of $\mathbf{S}$ obtained from SVD:

$$\mathbf{S} = \mathbf{S}_{col}\mathbf{\Lambda}\mathbf{S}_{row}^{T}. \tag{4.10}$$

The rotation matrix is used in each iteration of the EM algorithm to update the matrix $\mathbf{V}'$ and align it with $\mathbf{V}$,

$$\mathbf{V}' \leftarrow \mathbf{R}\mathbf{V}'. \tag{4.11}$$

Updating $\mathbf{V}'$ before estimating the statistical statistics of the latent variables, $\mathbf{y}_i$ and $\mathbf{w}_{ij}$, removes the redundancy in the second term of equation Eq. (4.6) by replacing the eigenvoice matrix $\mathbf{V}'$ with information obtained from the basis vectors in $\mathbf{V}$. Since both factors in Eq. (4.6) are guided towards representing the speaker space, the overall system achieves a better estimate of $\mathbf{V}$ and consequently more accurate estimates for latent variable statistics.

### 4.4.1 Co-channel Interference in Trials with dual eigenvoice PLDA

As described in Sect. 4.4, *dual eigenvoice PLDA* attempts to model a linear factorization of the i-Vectors into a target speaker and an interfering speaker component (see Eq. Eq. (4.6)). PLDA is able to distinguish the target speaker using the several recordings available for each speaker. The key difference between this method and *mixed PLDA* is that the system is forced to use a similar subspace to model the interfering speaker. Figure **??** shows the EER for different amounts of co-channel interference introduced to the trials.

Considering that *simplified PLDA* does not claim robustness towards co-channel interference, it shows little resistance as trial SIR values increase. Since *mixed PLDA* has some observation of the co-channel condition, it reduces the EER for at least 1% across all SIR conditions, as was shown in Sect. 4.3.1. *Dual eigenvoice PLDA*, further improves the performance and obtains an absolute of 2.5-3.5% drop in EER in all conditions. EER variations are

significantly different for *dual eigenvoice PLDA* model compared to the other two systems. Figure 4.8 shows that the clean-to-0$dB$ EER range for *mixed PLDA* is more than twice as much as *dual eigenvoice PLDA*.

### 4.4.2 Convergence of Dual eigenvoice PLDA

An interesting observation was made while developing *dual eigenvoice PLDA* regarding the convergence of model parameters, specifically the second eigenvoice matrix. Later, with some more investigation and feedback from experts in the field, I realized a significant flaw in the proposed formulation. This section provides a brief description of the issue with *dual eigenvoice PLDA*.

The claim in *dual eigenvoice PLDA* is that the first eigenvoice matrix in Eq. (4.3) contains factor loadings corresponding to the primary speaker and the second eigenvoice matrix (i.e., $\mathbf{V}'$) is for secondary speakers. We also chose a full-covariance slack variable, $\mathbf{z}$. The problem here arises from the full-covariance slack variable.

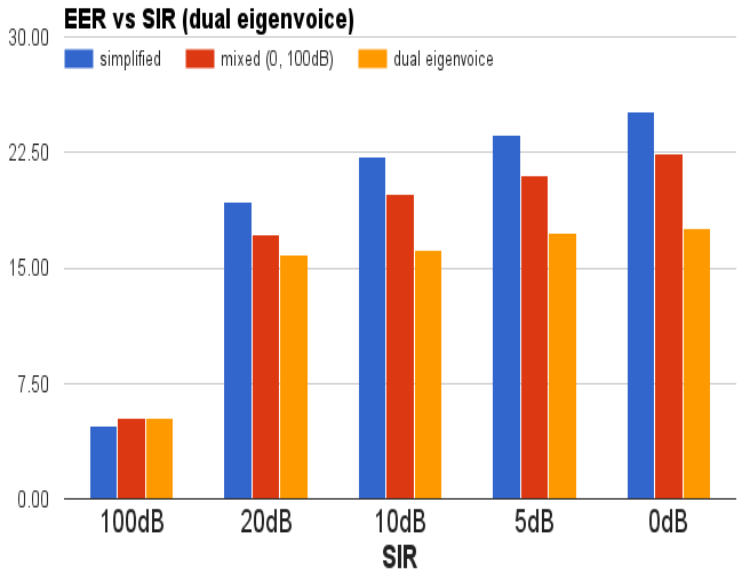


Figure 4.8. Comparing *dual eigenvoice PLDA*(yellow) with *mixed PLDA* (red) and *simplified PLDA* (blue). A steady improvement over *mixed PLDA* is observed across co-channel conditions.

## 4.5 Proposed method: Co-channel Aware PLDA

This section describes the last proposed approach, which are modifications to the PLDA covariances (Sect. 4.2) that allow modeling speaker interference in ways that are more appropriate for co-channel speech. In Sect. 4.4, a modified PLDA formulation was proposed to remove secondary speaker interference from the latent variable subspace (**?**). This section uses different methods based on the two-covariance interpretation (described in Sect. 4.2.3). The method, called *co-channel aware PLDA* (caPLDA), uses the dual distribution definition of Eq. (4.4) and Eq. (4.5) to model a co-channel i-Vector.

As mentioned before, caPLDA adopts the two-covariance interpretation of PLDA. The i-Vectors of a given class can be modeled as a normally distributed vector with a mean belonging to the class to which it belongs and a covariance matrix representing within class variability, Eq. (4.4). Typically, within-class variability is meant to model channel variation across sessions. In the case of co-channel speech, in addition to channel variability, one must consider the variability caused by interfering speakers. It was shown in Sect. 4.1.1 that speaker interference has a dramatic impact on error rates. Much more compared to channel variation in a dataset such as Switchboard. It is therefore reasonable to prioritize co-channel interference over channel mismatch.

In Sect. 4.3, we proposed capturing speaker interference in the same manner that channel variation is captured by PLDA (i.e., mixed PLDA). Although some improvement is attainable, we expect that the original PLDA formulation, Eq. (4.3), is not capable of fully capturing speaker interference; partly due to the similarity between speaker-dependent latent variables and the cross-session variations that exist in co-channel interference.

In order to improve performance, information can be shared from the between-speaker covariance to within-class covariances. For an i-Vector with speaker $A$ as the foreground (aka primary) speaker and some speaker $X$ as secondary speaker, PLDA assumes a normal distribution $\mathcal{N}(\mathbf{m}_A, \mathbf{\Phi}|A)$. While Eq. (4.4) considers $\mathbf{\Phi}$ to be a within-class covariance matrix

for all speakers in the i-Vector space (i.e., $\mathbf{\Phi}_w$), it can be argued that an additional component is required to model within-speaker variations in the case of co-channel i-Vectors. The additional component contributing to within-class covariance is of the same nature of the between-class covariance. Therefore, one can assume that $\mathbf{\Phi}$ is a function of both $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_b$, $\mathcal{F}(\mathbf{\Phi}_w, \mathbf{\Phi}_b)$. The suggested structure for $\mathcal{F}(.,.)$ is a linear combination of the two covariance matrices:

$$\Phi = \mathcal{F}(\mathbf{\Phi}_w, \mathbf{\Phi}_b) = \alpha_w \mathbf{\Phi}_w + \alpha_b \mathbf{\Phi}_b, \tag{4.12}$$

where $\alpha_w$ and $\alpha_b$ are functions of the signal-to-interference ratio between the foreground and background speaker.

Sections 4.5.1 and 4.5.2 visit different possibilities of $(\alpha_w, \alpha_b)$. However, we find it useful to first focus on a special case, which is to assume session variability should be of the exact same type as speaker variability in the case of co-channel i-Vectors. This special case assumes equal within- and between-speaker covariances.

### 4.5.1 Equal within- and between-speaker covariances ($\alpha_w = 0$)

In (Ioffe, 2006), PLDA model parameters are learned by maximizing the likelihood of observation i-Vectors provided for each speaker (i.e., class). The likelihood is calculated by assuming the i-Vectors are conditionally independent given the speaker. In this case, the probability of i-Vectors $\mathbf{m}^1, \mathbf{m}^2, ..., \mathbf{m}^n$ regardless of the speaker they were generated from, represented by $y$, is:

$$\mathcal{P}(\mathbf{m}^1\mathbf{m}^2...\mathbf{m}^n) = \int \mathcal{P}(\mathbf{y})\mathcal{P}(\mathbf{m}^1|y)\mathcal{P}(\mathbf{m}^2|y)...\mathcal{P}(\mathbf{m}^n|y)dy. \tag{4.13}$$

Replacing the probabilities using Eq. (4.4) and Eq. (4.5), the log-likelihood of $\mathcal{P}(\mathbf{m}^1\mathbf{m}^2...\mathbf{m}^n)$ is:

$$\mathcal{L}(\mathbf{m}^1\mathbf{m}^2...\mathbf{m}^n) = -\frac{C}{2}(ln|\mathbf{\Phi}_b + \frac{1}{n}\mathbf{\Phi}_w| + tr((\mathbf{\Phi}_b + \frac{1}{n}\mathbf{\Phi}_w)^{-1}\mathbf{S}_b) \\ +(n-1)ln|\mathbf{\Phi}_w| + ntr(\mathbf{\Phi}_w^{-1}\mathbf{S}_w)), \tag{4.14}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between-speaker and within-speaker scatter matrices calculated from the training i-Vectors. The variable $C$ is a normalizing constant. The log-likelihood is maximized using Eq. (4.14), resulting in Expectation-Maximization (EM) steps that estimate values for $\mathbf{\Phi}_w$ and $\mathbf{\Phi}_b$. If we ignore channel variability across sessions and assume that the only source of variation across different sessions is between-speaker variability, the probabilities $\mathcal{P}(\mathbf{x}_i|\mathbf{y})$ change from $\mathcal{N}(\mathbf{y}, \mathbf{\Phi}_w)$ to:

$$\mathcal{P}(\mathbf{x}_i|\mathbf{y}) = \mathcal{N}(\mathbf{y}, \Phi_b). \tag{4.15}$$

The assumption here is that within-session variation is similar to between-speaker variations (i.e., $\mathbf{\Phi}_b$). This assumption is equivalent to setting $\alpha_w$ to 0 and $\alpha_b$ to 1 in Eq. (4.12). The upside of setting $\mathbf{\Phi}$ to $\mathbf{\Phi}_b$ is twofold: 1) it increases the accuracy of estimating $\mathbf{\Phi}_b$, since in this scenario within-speaker differences can also be used to calculate $\mathbf{\Phi}_b$. 2) The maximum likelihood problem in Eq. (4.14) is reduced to:

$$\mathcal{L}(\mathbf{m}^1\mathbf{m}^2...\mathbf{m}^n) = -\frac{c}{2}(nln|\mathbf{\Phi}_b| + ln(\frac{n+1}{n}) + \frac{n}{n+1}tr(\mathbf{\Phi}_b^{-1}\mathbf{S}_b) \\ +ntr(\mathbf{\Phi}_b^{-1}\mathbf{S}_b)), \tag{4.16}$$

which is simpler to maximize. In fact, when all speakers have the same number of training i-Vectors ($n$), the solution is to set $\mathbf{\Phi}_b$ to $\mathbf{S}_b$.

### 4.5.2 Co-channel Interference in Trials with caPLDA

Finally, it is useful to analyze the potential gain in using alternative PLDA formulations described in Sect. 4.5. Instead of a grid-search of all possible $(\alpha_w, \alpha_b)$ inputs, only a selection of choices are presented here that we believe provide most useful insight.

Figure 4.9, shows the case in which $\alpha_w$ is set to 1 and $\alpha_b$ is varied from 0 to 1. The EER values are almost unequivocally lower than their counterparts presented in Table 4.1, despite the fact that they do not use mixed PLDA training data. In other words, the EER values calculated in Fig. 4.9 show performance using clean ($100dB$) PLDA training data. The bottom line (blue) shows EER for clean trials. As expected, EER values slightly increase in the clean condition, when the original PLDA formulation is modified. However, in lower SIR conditions ($20 - 0dB$), more performance gain is observed as $\alpha_b$ approaches 0.1.

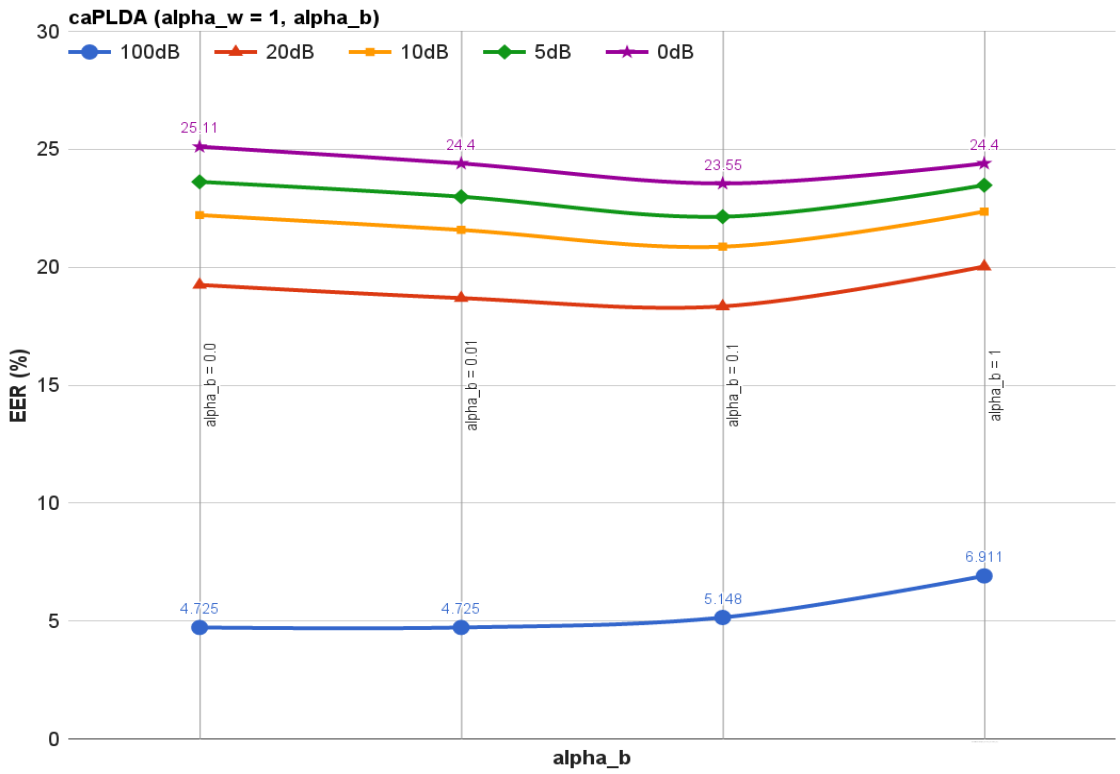


Figure 4.9. Comparing caPLDA for different values of between-speaker coefficient ($\alpha_b$). Here we set $\alpha_w$ to 1. The curves start from left with $\alpha_b = 0$, the scenario equivalent to what is shown in Fig. 4.5. The bottom (blue) lines shows the effect on clean trials, in which modifying simplified PLDA always degrades performance. For co-channel trials, however, setting $\alpha_b$ to non-zero values improves performance for all SIR values.

The second attempt is to investigate the special case in which $\alpha_w$ is set to 0 and $\alpha_b = 1$, as proposed in Sect. 4.5.1. We argued that in this scenario the estimate for $\Phi_b$ is likely to be most accurate. Table 4.2 compares using the three mixed PLDA scenarios plus clean

PLDA. From the last two columns, mixed PLDA $(0, 100dB)$ and clean PLDA, we see that performances are fairly close despite the additional training data in mixed PLDA $(0, 100dB)$.

Table 4.2. caPLDA ($\alpha_w = 0$,$\alpha_b = 1$) using different co-channel training conditions. The last column show performance without co-channel data in PLDA training.

| SIR (dB) | caPLDA + mixedPLDA(0dB) | | caPLDA + mixedPLDA(0,5,10,20,100dB) | | caPLDA + mixedPLDA(0,100dB) | | caPLDA | |
|---|---|---|---|---|---|---|---|---|
| | EER (%) | minDCF | EER (%) | minDCF | EER (%) | minDCF | EER(%) | minDCF |
| 100 | 9.03 | 0.648 | 7.12 | 0.554 | **6.63** | 0.526 | **5.71** | 0.465 |
| 20 | 21.16 | 0.925 | 18.55 | 0.890 | **18.48** | 0.884 | **18.83** | 0.915 |
| 10 | 23.34 | 0.956 | 20.94 | 0.940 | **20.52** | 0.929 | **21.65** | 0.953 |
| 5 | 24.96 | 0.971 | 22.21 | 0.957 | **22.00** | 0.951 | **22.57** | 0.975 |
| 0 | 26.09 | 0.983 | 23.62 | 0.978 | **23.06** | 0.970 | **23.70** | 0.986 |

## 4.6 Summary

From this chapter, advancements with improved PLDA for co-channel speech carry most of the weight in terms of novelty for this dissertation. As mentioned in Chapters 1 and 4, the entire perspective of separating overlapped speech from co-channel is a missing concept in existing literature. It was argued in Section 4.1 that in conventional conversational English, the detrimental impact of co-channel interference is an order of magnitude greater than the impact of overlap. This was shown over a number of experiments on existing large-scale conversational datasets.

The second part of this chapter provides solutions to the problem of co-channel speech in the latent variable space (i.e., i-Vector sub-space). The proposed methods were: 1) mixed-PLDA, which is considered the standard way of compensating mismatch using probabilistic linear discriminant analysis (PLDA); 2) dual-eigenvoice PLDA, which considers adding a second eigenvoice matrix to the PLDA formulation; 3) co-channel aware PLDA, which uses the two-covariance interpretation of PLDA to adjust/modify covariance matrices in PLDA. Each of these methods shows improvement from a different perspective. MixedPLDA shows stable performance moderate improvement over all signal-to-interference conditions, but requires adequately tuned co-channel development data. Alternatively, dual-eigenvoice PLDA

shows superior performance and does not require tuned parameters, co-channel aware PLDA is more stable and does not suffer from singularities during the training process.

# CHAPTER 5

# SPEAKER DIARIZATION IN CO-CHANNEL SPEECH

Throughout the course of this study, part of the agenda has been to acknowledge non-overlapping speech as an important component of co-channel data in single-channel audio streams. It was shown that distinguishing overlap from co-channel introduces more realistic scenarios to the scope of co-channel speech problems. An important example of such problems is conversational speech, a common and realistic form of speech. Among the various aspects of analyzing conversations, *speaker diarization* is closest to speaker recognition.[1] Speaker diarization refers to the task of automatically determining "who spoke when?" within an audio signal containing two or more speakers. This chapter addresses the tasks of 1) segmenting co-channel data into single-speaker excerpts and 2) clustering segments, which means to group them by speakers. Tasks are described within the context of CRSS-SpkrDiar, a speaker diarization tool-kit designed to perform diarization while simultaneously supporting speaker recognition and speech recognition using Kaldi (Povey et al., 2011). Therefore, the contribution of this chapter is to:

- introduce an end-to-end conversation analysis research platform that is tightly connected to Kaldi and does not require cross-platform coding interfaces.

CRSS-SpkrDiar is a speaker diarization tool-kit developed as part of a collaboration with another student, Chengzhu Yu, a fellow PhD student at the Center for Robust Speech Systems (CRSS). The main motivation behind developing this tool-kit is to establish an integrated end-to-end conversation analysis system that provides the capability of diarizing signals while supporting speech recognition. Currently, one of the most popular speech recognition platforms used in research is Kaldi, developed in Johns Hopkins University (Povey

[1] As a reminder for the readers, speaker recognition is the main theme of this thesis.

et al., 2011). Existing diarization systems are implemented in different platforms and to the best of our knowledge none support Kaldi I/O functions. Switching between platforms and APIs (Application Programming Interface) frustrates users who are interested in simultaneously analyzing speaker diarization and speech recognition. We hope that developing a speaker diarization module that employs Kaldi will help students and staff with the kind of multi-purpose research that is common in CRSS. Although CRSS-SpkrDiar is currently in working condition, as a research platform it is always considered under development and is available for those who are interested in investigating various aspects of conversational data.

In Sect. 5.1, a layout of the system and its relationship with Kaldi is presented. Section 5.1 also provides a list of our modules and their Input/Output. Here, we also explain how these modules interact with Kaldi data types. In Sect. 5.2, segmentation, the first task of speaker diarization, is described. The purpose of segmentation is to split signals into smaller chunks that contain only one speaker. Some segments may contain overlapped speech or no speech at all. Therefore, as part of segmentation, speech activity detection (SAD) and overlap detection modules are also integrated into the system. In Sect. 5.3, the clustering (grouping) module is presented. A state-of-the-art technique, called integer linear programming (ILP), is used in CRSS-SpkrDiar to cluster segments obtained from the segmentation stage (Bredin and Poignant, 2013). Clustered groups represent segments that belong to the same speaker. Another component also described in Sect. 5.3 is the re-segmentation, which acts as a correction layer on top of the clustering module. Sections 5.4 and 5.5 describe some of the novel techniques used to improve clustering performance in CRSS-SpkrDiar. Section 5.6 summarizes some evaluations on the AMI meeting corpus. Finally, Sect. 5.7 points out some of the future work required to further improve CRSS-SpkrDiar.

Part of the reason this chapter is placed at the end of the dissertation is that it can be viewed as a description of a comprehensive system that contains all problems considered in this thesis. Furthermore, since CRSS-SpkrDiar is a research platform that could potentially

benefit those looking beyond the scope of this study, it is considered a gateway to future work beyond this thesis.

## 5.1 CRSS-SpkrDiar Layout

The approach in speaker diarization is to split an audio recording (usually at least a few minutes long) into smaller segments that only contain one speaker. After completing the first step, one must label the segments according to speaker identities. The assumption is that no prior knowledge of the number of speakers or their identities is available. The only possible solution, therefore, is to compare the segments and group those that appear to belong to the same speaker.

The segment-and-cluster approach is the most common solution to speaker diarization (Meignier and Merlin, 2010; Anguera et al., 2012). That being said, alternative approaches have been proposed, especially with regard to segmentation and post-clustering steps. Some studies completely ignore the segmentation step and use equal-length segments to perform clustering (Sell and Garcia-Romero, 2015). Skipping segmentation is an attractive proposition, since as will be shown in the following section, segmentation is prone to errors. In addition, using equal lengths instead of varying-length segments to some extent guarantees equal reliability of individual segments for the clustering step. We say equally reliable because speaker-dependent features (e.g. i-Vectors) are known to depend largely on the length of the signals from which they are extracted (Hasan et al., 2013). When automatic segmentation is used to split the audio recording, segment lengths may vary causing the corresponding speaker-dependent features to vary in quality and reliability.

Another component in speaker diarization, not mentioned above, is the re-segmentation step. Re-segmentation is a post-clustering step used to prevent sudden changes in the speaker. The reason this module is important is the fact that clustering does not take time sequences into account when grouping segments. Meanwhile, as listeners we expect a

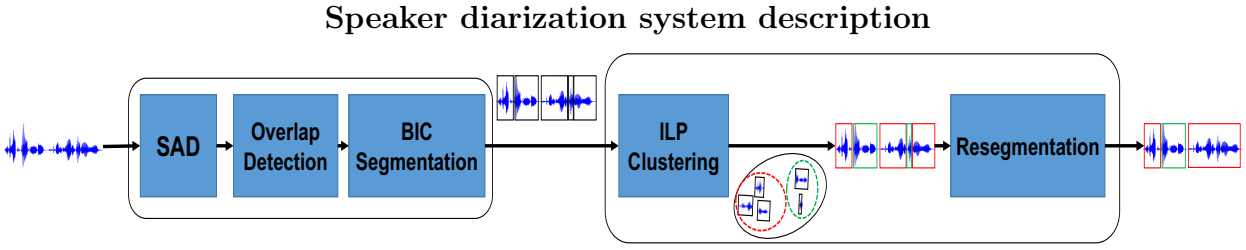


Figure 5.1. CRSS-SpkrDiar system overview. Two main steps are used in speaker diarization: 1) Segmentation (SAD, overlap detection, and BIC segmentation) and 2) clustering (ILP clustering and resegmentation).

certain amount of coherence in conversations. This presumed coherence in the sequence of speakers is used to correct errors in the clustering step.

Figure 5.1 shows an overview of CRSS-SpkrDiar. The input is the audio recording of two or more speakers. We start by performing speech activity detection (SAD) and overlap detection to label non-speech and overlapped segments as off-bounds. The next step is segmentation, which uses a measure called Bayesian Information Criterion (BIC) to detect speaker change points (Chen and Gopalakrishnan, 1998; Zhou and Hansen, 2005). After BIC-Segmentation, we have access to segments within the audio recordings and would like to label these segments to determine which belongs to which speaker (speaker A, B, C, etc.). Of course, these speaker identities are assigned relative to the input audio and are not actual speaker identities. This means that given another input signal, the speaker diarization system will provide similar labels, but speaker A in audio 1 has no relation to speaker A in audio 2. CRSS-SpkrDiar generates speaker labels using integer linear programming (ILP) to cluster the segments. ILP, described in detail in Sect. 5.3, uses a global optimization approach to minimize speaker diversity within each cluster while simultaneously minimizing the number of clusters (i.e., speakers).

Thus is the overview of a standard speaker diarization system. As mentioned before, there are many implementations available for speaker diarization. What CRSS-SpkrDiar offers in addition is to allow speaker diarization in a platform that also supports speech and speaker recognition (Kaldi).

### 5.1.1 Interaction with Kaldi

As pointed at the beginning of the chapter, a driving force in developing this tool-kit was high compatibility with the Kaldi speech and speaker recognition platform (Povey et al., 2011). For those not familiar with the Kaldi project, it is a toolkit for speech recognition written in C++ and licensed under the Apache License v2.0. Kaldi is intended for use by speech recognition researchers (Povey et al., 2011). Kaldi comprises many modules each designated to a specific task. For example, *feat* for feature extraction or *ivector* for ivector-based speaker recognition. Most modules come with a corresponding *bin* directory that contains executable files used to perform various functions in a speech or speaker recognition system (e.g. *featbin*, *ivectorbin*). The executables are those most users interact with in their Bash scripts, which are also referred to as "recipes".

CRSS-SpkrDiar follows the same convention and consists of a *diar* and a *diarbin* directory. The classes are defined in *diar*, while *diarbin* contains the executables used to perform segmentation and clustering. CRSS-SpkrDiar uses matrix and utility libraries from Kaldi in its core. Kaldi also includes modules that define Gaussian mixture models and Hidden Markov models (*gmm* and *hmm*), of which we also take advantage for acoustic modeling. As mentioned in the previous section, i-Vectors are used in CRSS-SpkrDiar as speaker-dependent features. Therefore, Kaldi's *ivector* is used to extract i-Vectors and calculate distances and models such as PLDA (probabilistic linear discriminant analysis) to perform clustering. Figure 5.2 shows the interaction between *diar* and *diarbin* components and Kaldi libraries.

## 5.2 Segmentation

This section briefly describes the layout of the Segmentation component in CRSS-SpkrDiar. Traditionally, the approach for speaker diarization has been to first identify speaker change

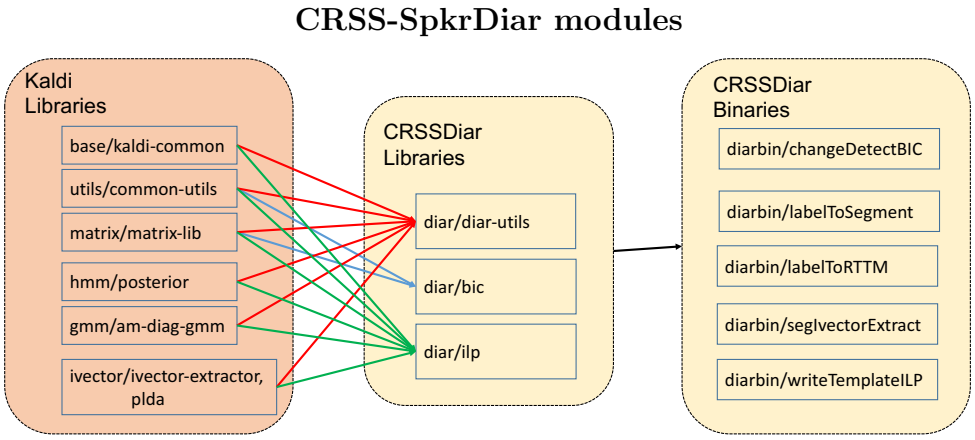


Figure 5.2. CRSS-SpkrDiar components and their relation with Kaldi libraries.

points in an audio stream. This is due to the restrictive framework of typical speaker diarization problems. The restriction being no prior knowledge of the number or identity of the speakers in the signal. Therefore, the solution is to track acoustic features along the signal and detect points at which these features demonstrate significant changes.

Segmentation in the context of this study is formally defined as detecting the time indexes corresponding to changes in an ordered sequence of acoustic features (Cettolo et al., 2005). One of the most popular speaker change detection algorithms used in speaker diarization is the Bayesian information criterion (BIC), which is briefly described in this section. It should be pointed out that BIC segmentation has been thoroughly studied in the research community and several resources are available that substantially cover the essentials required to understand its function in a speaker diarization system (Cettolo and Vescovi, 2003; Cettolo et al., 2005; Chen and Gopalakrishnan, 1998; Zhou and Hansen, 2005). The content here is intended to describe all aspects of CRSS-SpkrDiar and share implementation details for the curious reader.

### 5.2.1 Bayesian Information Criterion

The Bayesian information criterion (BIC) is a measure that determines the effectiveness of a given probabilistic model in capturing statistical characteristics of a set of data samples.

BIC is used in segmentation as a way to quantify the certainty of the existence of a break-point in a given window of the audio. Therefore, instead of finding a global solution to the segmentation problem defined in Sect. 5.2, the standard BIC implementation tracks a sliding window in the audio and compares the following two hypotheses:

- $H_0$: The window does not contain a breaking point.
- $H_1$: The window contains exactly one breaking point.

In speaker diarization terms, $H_0$ implies that the window contains one speaker, while $H_1$ implies that two speakers exist. BIC segmentation tests these hypotheses by using the following function in a sequence of acoustic features in a window, $\{X_1X_2...X_N\}$, for the hypothesis $H$:

$$BIC(H) = log\mathcal{L}_M(H) - P_{M_N}, \tag{5.1}$$

where $\mathcal{L}_M(H)$ is the maximum likelihood of the model $M$ under hypothesis $H$. $P_M$ is the BIC penalty function (Schwarz et al., 1978) when $M$ is used to parameterize $N$ samples. The indexes correspond to time frames.[2]

$$P_{M_N} = \frac{F_M}{2} log(N), \tag{5.2}$$

with $F_M$ as the degrees of freedom of the chosen model, $M$. When the sequence $\{X_1X_2...X_N\}$ comprises two segments (i.e., $H = H_1$), the break-point is assumed to be an index $i_{max}$ between 1 and $N$. $BIC(H_1)$ is maximized at the break-point $i = i_{max}$. This assumption states that breaking the sequence into two sub-sequences $\{X_1X_2...X_{i_{max}}\}$ and $\{X_{i_{max}+1}X_{i_{max}+2}...X_N\}$ maximizes the BIC penalty. On the other hand, if the sequence does not contain a break-point (i.e., $H_0$), $i_{max}$ can be assumed equal to $N$. The actual measure used in BIC segmentation is the difference between the BIC corresponding to $H_0$ and the BIC of $H_1$. This value

[2]As we know, acoustic features, $X_i$, are calculated over typically $25msec$ frames. The framing used throughout this chapter uses $25msec$ frames with an increment of $10msec$. These frames are not to be mistaken with windows used to search for change points.

is called $\Delta BIC_i$:

$$\Delta BIC_i = (log\mathcal{L}_{M_i} - P_{M_i}) - (log\mathcal{L}_{M_N} - P_{M_N}) \tag{5.3}$$

where $\mathcal{L}_{M_i}$ is short-hand for the maximum likelihood of $H_1$ when the break-point is at index $i$. If $\Delta BIC_i > 0$, the model favors $H_1$ at breaking point $i$, and otherwise the model suggests no break-points ($H_0$).

$\Delta BIC$ depends on the model chosen for $M$. A common choice in speaker diarization is a Gaussian model, which assesses whether the sequence is best described with one covariance matrix ($H_0$) or two ($H_1$). In (Cettolo et al., 2005), $\Delta BIC_i$ is derived for Gaussian modeling with the inclusion of a sensitivity factor $\lambda$.

$$\Delta BIC_i = \frac{N}{2}log|\Sigma| - \frac{i}{2}log|\Sigma_1| - \frac{N-i}{2}log|\Sigma_2]| - \lambda P_{M_N} \tag{5.4}$$

where $\Sigma$ is the maximum likelihood estimate of the covariance matrix of $\{X_1X_2...X_N\}$. Similarly, $\Sigma_1$ corresponds to the sequence $\{X_1X_2...X_i\}$ and $\Sigma_2$ to $\{X_{i+1}X_{i+2}...X_N\}$. $P_M$ in Eq. (5.4) is the BIC penalty of modeling $N$ samples with a Gaussian. The number of free parameters in a multi-variate Gaussian is $d + \frac{d(d+1)}{2}$, in which $d$ is the dimension of the acoustic features. The degrees of freedom include $d$ values for the Gaussian mean and $\frac{d(d+1)}{2}$ values for the symmetric covariance matrix.

In addition to identifying whether a window contains one or two speakers, the segmentation module is also expected to return a time index for the change point. BIC segmentation returns this value by incrementally increasing the index $i$ until the point at which $\Delta BIC_i$ is positive. Once the change point is detected, granted that it exists in the first place, a recursive search is conducted with smaller increments to find the exact change point.

Now that the background theory of BIC segmentation has been presented, we move on to a description of BIC segmentation in CRSS-SpkrDiar in a module called change-point detection.

### 5.2.2 Change-Point Detection in CRSS-SpkrDiar

Change-point detection, or change detection in short, is a tool used to estimate segment boundaries in an audio stream. It performs BIC segmentation on speech segments produced from speech activity and overlap detection. From here on speech segments refer to segments that may contain multiple non-overlapping speakers or a single speaker. The no-overlapping property is implied unless "overlapped segments" is used. The goal is to split segments at locations where speakers change. There is no prior assumption regarding the number of speakers in any given segment. The output of change detection, however, will ideally produce smaller segments that each contain only one speaker.

Given a segment, change detection first examines whether the segment is long enough to compute reliable BIC estimates (as defined in Sect.5.2.1). For segments that are sufficiently long, a sliding window is defined that incrementally grows in size. This window will grow until: 1) either a change point is detected using the $\Delta BIC$ measure from Eq. (5.4) or 2) the length of the window reaches a maximum acceptable length at which point the window is assumed to contain only one speaker (no change points are detected). As mentioned, the window slides until it reaches the end of the segment and returns the change points that were found during the search.

Every time a change point is detected the window first performs an identical search with more refined search increments until it finds the feature index, $i$, at which $\Delta BIC_i$ is maximized. After finding $i_{max}$, the window is reset and continues the search at $i_{max} + 1$.

Figure 5.3 shows the pseudo-code for change detection implemented in CRSS-SpkrDiar. The algorithm requires a set of user defined parameters that determine the search speed and BIC sensitivity.

- $N_{min}$: minimum window length (in number of frames) used to start search for change point. Segments shorter than this value are not long enough to estimate $\Delta BIC$.

- $N_{max}$: maximum window length. The search window grows incrementally until it reaches this maximum length. If the search window reaches this value before a change point is detected, the algorithm quits the search and declares no change point.

- $N_{margin}$: window margin. These margins are used to assure that sufficient number of frames are available to compute BIC estimates.

- $N_{second}$: refining window length. Once a change-point is detected, the search is refined using a smaller window of this length.

- $N_{shift}$: sliding parameter. The window slides to the right once its length reaches $N_{max}$.

- $N_{grow}$: The incremental growth factor. Every time a window is examined for change points, the window length increases by $N_{grow}$.

- $\lambda$: BIC sensitivity factor. The higher this value, the more sensitive $\Delta BIC$ is to changes.

- *lowResolution*: initial search increment used to calculate $\Delta BIC_i$. The value $i$ increments by *lowResolution* in each iteration.

- *highResolution*: refined search increment. After a change point is detected in the initial search, a second search window is created and search with an increment of *highResolution*.

An important component of change detection is the *segment* data type. This class contains all the necessary information to calculate functions used in speaker diarization. For example, each *segment* contains two variables, *Begin* and *End* that indicate the start and end frames corresponding to that segment. This class also includes functions used to calculate i-Vectors, which will be used extensively in the next section. *Segments* is considered the core data type used in CRSS-SpkrDiar and is highly compatible with existing Kaldi classes.

```
SplitSegment
// Perform BIC change dectect on single speech segment.
if (Nmin >= segment.Length)
   return
window.Create(segment.Start + Nmargin,Nmin)

while (window.End <= segment.End)
   maxBICIndexValue = DeltaBIC(window, lowResolution)
   while (maxBICIndexValue <= 0 & window.Length < Nmax) & window.End <=
       segment.End)
      window.GrowWindow(Ngrow)
      maxBICIndexValue = DeltaBIC(window,lowResolution)

   while (maxBICIndexValue <= 0 & window.End <= segment.End)
      window.ShiftWindow(Nshift)
      maxBICIndexValue = DeltaBIC(window,lowResolution);

if (maxBICIndexValue.second > 0 & window.End <= segment.End)
   window.Restart(Nsecond)
   maxBICIndexValueHighRes = DeltaBIC(window, highResolution)
   if (maxBICIndexValueHighRes.second > 0)
      i_max = maxBICIndexValueHighRes.Index
      bicSegments.Create([segment.Begin,i_max]);
      segment.Begin = i_max + 1;
      window.ResetWindow(segment.Begin,Nmin);
else
   window.ResetWindow(segment.End - Nmargin + 1, Nmin)
```

Figure 5.3. Algorithm - change detection using BIC. The algorithm describes the two-step procedure of searching for change points in an audio segment. The first step is to search for change points using larger increments, *lowResolution*. Once the change point is detected, a second, more refined, search is performed in a smaller search window using *highResolution*.

## 5.3 Clustering

This section describes the second step in speaker diarization, clustering the segments. After single-speaker segments are identified, the speaker diarization system must decide which segments belong to the same speaker. This is accomplished without prior knowledge of the number of speakers or their identities. The task of grouping segments that belong to the same speaker is known as clustering.

A number of bottom-up and top-down solutions have been suggested for the clustering step in speaker diarization (Tranter and Reynolds, 2006; Anguera et al., 2012). Top-down approaches start by assuming that all segments belong to a single cluster and iteratively split clusters into smaller, more exclusive, groups. Bottom-up clustering, on the other hand, starts by assuming each sample is a separate cluster and iteratively merges clusters into more inclusive groups. The most popular of these is a bottom-up algorithms called hierarchical agglomerative clustering (HAC). This approach first assumes that all segments belong to a separate cluster (i.e., speaker) and iteratively merges the two closest clusters. A similarity criterion is used to measure the distance between clusters. Performance varies depending on the criterion used (e.g., BIC, Kullback-Leibler divergence). This iterative process continues until the similarity criterion is no longer satisfied between any two clusters. As opposed to the single multivariate Gaussian used to detect change points in the previous section, the clustering process usually uses more sophisticated measures. Gaussian mixture models (GMM) have been proven effective in modeling clusters (Zelenák et al., 2010). As clusters merge, the amount of data available to estimate GMM parameters increases. Therefore, GMM reliability increases in HAC iterations. It is easy to see why using GMMs was not feasible in the segmentation step, since the amount of data is too small to model an entire GMM.

Despite its popularity, HAC is associated with a number of issues. The most important being the error-propagation that occurs during iterations. If two segments are incorrectly

grouped together in one iteration, the model (e.g., GMM) used to represent the cluster for the next iteration will not accurately represent the speaker identity, and so on. This issue has led many to use other top-down clustering approaches, which are less likely to suffer from error-propagation. Examples being K-means (Shum et al., 2011) or spectral clustering (Shum et al., 2012; Ning et al., 2006). To the best of our knowledge, there are no conclusive results that suggest one clustering method over all others for speaker diarization. Another important development with regard to clustering has been to use i-Vectors to perform clustering. The standard HAC-GMM framework does not use i-Vectors in the clustering process. Some developments have been made to use i-Vector distances in an HAC clustering solution. The reader should be reminded that the compatibility of CRSS-SpkrDiar with Kaldi allows for a complete integration of i-Vectors with clustering.[3]

An increasingly popular algorithm that has been used in speaker diarization has been *integer linear programming* (ILP). Two reasons that make ILP particularly interesting are:

- Given appropriate formulation, ILP can result in global optimum for some clustering problems.
- A vast bed of research exists for linear programming algorithms. Integrating these solutions into any problem, in this case speaker diarization, provides new insights and advantages.

ILP was introduced as a global optimization solution for the clustering problem in speaker diarization by Rouvier and Meignier (Rouvier and Meignier, 2012). An attractive aspect of ILP for speaker diarization is that it is relatively strait-forward to perform ILP clustering on i-Vectors (Dupuy et al., 2012). The implementation in CRSS-SpkrDiar is based on improvements on the original ILP formulation for speaker diarization (Dupuy et al., 2014).

[3]While this manuscript was under preparation, HAC was also added to CRSS-SpkrDiar. The most current version of the software supports both HAC and ILP clustering methods.

The single-speaker segments from change detection can be used to derive i-Vectors. As you may recall from Chapter 4, i-Vectors are fixed-length vectors that represent speaker-specific characteristic of a given audio signal of variable length. ILP is a constrained optimization problem that determines which i-Vectors should fall in the same cluster. The optimization ensures that:

1. all i-Vectors in a cluster are within a predefined distance, $\delta$, of all the other i-Vectors in that cluster.

2. the number of clusters is minimal.

3. every i-Vector is assigned to one and only one cluster.

The inherent compatibility of ILP allows formulating the aforementioned conditions in a convex linear programming manner, regardless of the non-linearities of distance metrics. The linearity comes from the fact that distance measures are assumed pre-calculated and do not depend on the choice of clusters. It was mentioned before that in AHC, the choice of cluster members in each iteration modified the distances, since it affected the cluster model for the next iteration. The same applies to top-down clustering, such as K-means. However, distances in ILP are constant and do not depend on the cluster labels. Therefore, ILP claims to provide a global minimum to the following equation derived form conditions (1) and (2) from the list above:

for clusters $C \in \{1...N\}$:

$$minimize \quad \frac{1}{\delta}\sum_{i=1}^{N}\sum_{j=1}^{N} dist(i,j)x_{i,j} + \sum_{k=1}^{N} x_{k,k} \tag{5.5}$$

The minimization problem searches for elements of a binary $N \times N$ clustering assignment matrix, in which $N$ corresponds to the number of i-Vectors (i.e., segments) generated in Sect. 5.2. The value $x_{i,i}$ is 1 if the $i^{th}$ i-Vector is identified as a cluster centroid and 0

otherwise. $x_{i,j}$ is 1 if the $j^{th}$ i-Vector belongs to the cluster whose centroid is $i$. The first term in Eq. (5.5) makes sure that centroids are picked in a way that minimizes the sum of distances of all elements in a cluster from their assigned centroid. The distance between the $i$ and $j^{th}$ i-Vectors is $dist(i,j)$. Clearly, the optimal solution to only minimize the first term, $\frac{1}{\delta}\sum_{i=1}^{N}\sum_{j=1}^{N} dist(i,j)x_{i,j}$, is to choose all i-Vectors as a centroid, which will undesirably result in $N$ clusters. Hence, the second term is introduced to simultaneously minimize the number of clusters. $\sum_{k=1}^{N} x_{k,k}$ is the sum of diagonal elements, also equal to the number of clusters. The first term is normalized by a factor $\delta$. This normalizing factor is the threshold used as the radius for all clusters. To assure that all clustering conditions are satisfied, a set of constraints are added to Eq. (5.5). Some conditions are implied by the formulation, such as:

$$x_{i,j} \in \{0,1\}, \qquad 1 \leq i,j \leq N \tag{5.6}$$

which states that the clustering assignment matrix must have binary elements; hence the phrase **integer** linear programming. Some constraints are straight-forward, for example each i-Vector can only be assigned to one cluster:

$$\sum_{i=1}^{N} x_{i,j} = 1, \qquad 1 \leq i \leq N \tag{5.7}$$

or that all distances should be less that a pre-determined threshold, $\delta$:

$$d(i,j)x_{i,j} < \delta. \qquad 1 \leq i,j \leq N \tag{5.8}$$

A more subtle constraint is that an i-Vector $j$ is assigned to a cluster only if the cluster has a centroid. In other words, $j$ can be assigned to $k$ if $k$ is a centroid.

$$x_{k,j} - x_{k,k} \leq 0. \qquad 1 \leq k,j \leq N \tag{5.9}$$

The minimization problem can be reformulated into a more compact problem (Dupuy et al., 2014). Using compact solutions with more restrictive constraints reduces the number

of variables that need to be solved in the linear programming problem. The following formulation increases the speed of the ILP solver by limiting the number of restrictive variables in Eq. (5.5). For $j \in \{1...N\}$ let $K_j = \{k|d(k,j) < \delta\}$

$$\begin{aligned} minimize \quad & \frac{1}{\delta}\sum_{j=1}^{N}\sum_{k\in K_j} dist(k,j)x_{k,j} + \sum_{k=1}^{N} x_{k,k} \\ subject\ to: \quad & x_{k,j} \in 0,1, && k \in K_j, 1 \leq j \leq N \\ & \sum_{k\in K_j} x_{k,j} = 1, && 1 \leq j \leq N \\ & x_{k,j} - x_{k,k} < 0. && k \in K_j, 1 \leq j \leq N` \end{aligned} \tag{5.10}$$

The optimzation problem in Eq. (5.10) is implemented in CRSS-SpkrDiar using the GNU linear programming kit (GLPK) (Makhorin, 2008). GLPK is an open-source C++ library (As mentioned before, the entire CRSS-SpkrDiar module solely depends on C++ libraries). GLPK explicitly reads the objectives, equalities, and inequalities in the form provided in Eq. (5.10). To create the input file for GLPK, i-Vectors are extracted from the segments and indexed appropriately. A distance matrix (see Sect.5.4 and 5.5) is calculate using all segment i-Vectors. Figure 5.4 summarizes the process of generating the GLPK input, we is called the *GLPK Template* in CRSS-SpkrDiar.

The descriptions provided in this section were based on existing studies (Mueller and Kramer, 2010; Rouvier and Meignier, 2012; Dupuy et al., 2012, 2014). What was presented here was a brief of summary of existing publications. A novel addition in CRSS-SpkrDiar has been in the choice of distance metric, $dist(.,.)$, described in the next section.

## 5.4 PLDA in ILP clustering

Probabilistic linear discriminant analysis (PLDA) was thoroughly investigated in Chapter 4 as a way to determine whether two individual i-Vectors (or two groups of i-Vectors) belong

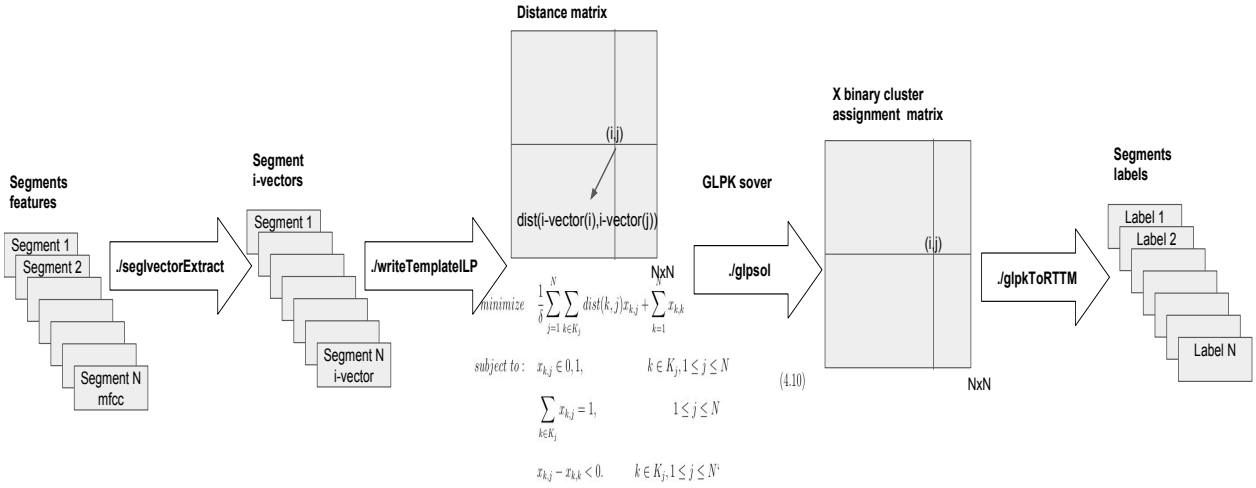


Figure 5.4. Summary of ILP clustering using CRSS-SpkrDiar. The binary executables used for each step are shown in the figure.

to the same speaker. The same question is faced during the clustering process for speaker diarization. Since ILP does not add any restrictions on the distance measures (see Eq. (5.10)), PLDA can be used here as the distance metric, $dist(.,.)$. This creates a more appropriate setting for the clustering step, which aims to compare speaker identities across segments.

The use of PLDA in speaker diarization has been proposed in a number of studies (Prazak and Silovsky, 2011; Silovsky et al., 2011; Sell and Garcia-Romero, 2014). PLDA proves particularly useful in a class of speaker diarization problems that perform cross-session diarization. Cross-session diarization simultaneously clusters segments from several audio streams. In this scenario PLDA functions as a channel compensation algorithm. CRSS-SpkrDiar proposes to use PLDA as a distance measure to perform global optimization in ILP clustering. This is expected to improve performance relative to Mahalanobis and cosine distances previously proposed for ILP clustering (Dupuy et al., 2012, 2014).

CRSS-SpkrDiar accepts labeled background i-Vectors to train a PLDA model (using Kaldi's PLDA class) to calculate distances. The function *computeDistanceMatrix* calculates PLDA distances from segment i-Vectors and background i-Vectors in *writeTemplateILP*. In

addition to *computeDistanceMatrix* for PLDA, CRSS-SpkrDiar also calculates cosine distance, Mahalanobis distance, and conditional Bayes distance. Section 5.5

## 5.5 Other distance measures

In addition to PLDA, three other distance measures are implemented for the clustering procedure in CRSS-SpkrDiar.

The first is cosine distance, which uses the normalized dot product of i-Vector pairs to define similarity. The cosine distance between two i-Vectors $\mathbf{v_i}$ and $\mathbf{v_j}$ is:

$$dist_{cos}(i,j) = \frac{< \mathbf{v_i}, \mathbf{v_j} >}{||\mathbf{v_i}||||\mathbf{v_j}||} \tag{5.11}$$

Cosine distance in its raw format defined in Eq. (5.11) is slightly incompatible with the distance metric required in Eq. (5.10). The problem is that higher cosine distance is associated with closer vectors. Also, $dist_{cos}$ allows negative values. ILP minimization, on the other hand, considers lower distance values (starting from 0) as close vectors and higher values (always positive) as distant. Therefore, the cosine metric used in CRSS-SpkrDiar is actually $1 - dist_{cos}(i,j)$.

The second distance metric is Euclidean distance, defined as the norm-2 distance between $\mathbf{v_i}$ and $\mathbf{v_j}$. Similarly, a Mahalanobis distance, which is a weighted Euclidean distance is also applicable. The weight for Mahalanobis distance is calculated from the covariance estimate of all the i-Vectors obtained from segments, $\mathbf{\Sigma}$.

$$dist_{mah}(i,j) = \sqrt{(\mathbf{v_i} - \mathbf{v_j})^T \mathbf{\Sigma}^{-1} (\mathbf{v_i} - \mathbf{v_j})} \tag{5.12}$$

For Euclidean distance $\mathbf{\Sigma} = \mathbf{I}$.

The last distance measure is conditional Bayes. This distance was suggested by Rouvier and Meignier (Rouvier and Meignier, 2012) as a substitute for Mahalanobis distance. Instead of calculating $\mathbf{\Sigma}$ from the segment i-Vectors, labeled development i-Vectors are used from

an external dataset. This measure computes a Mahalanobis distance using the within-class covariance corresponding to development speakers. The development data contains groups of i-Vectors from different speakers, for which speaker labels are available. The within-class covariance is calculated by averaging the covariance obtained from the i-Vectors of each development speaker. The implied assumption here is that all speaker classes have a similar within-class covariance. The goal of conditional Bayes is to remove channel-dependent variabilities from the i-Vectors.

## 5.6 Evaluation

This section shows preliminary results from CRSS-SpkrDiar. The experiments are conducted on the AMI meeting corpus, briefly described in Chapter 4. AMI consists of audio recordings from meetings, typically $30min$ sessions. The corpus also provides additional background information, including individual speakers' speech recorded from head-set and lapel microphones. In addition to audio files, the corpus contains segmentation ground-truth for speaker diarization. CRSS-SpkrDiar uses this information to calculate diarization error rates (DER) for each session separately. The way CRSS-SpkrDiar addresses different datasets is very similar to Kaldi. In that CRSS-SpkrDiar binaries are independent of the data-set, but users can create recipe scripts that are specific to a particular corpus design.

### 5.6.1 Diarization Error Rates

The most popular error rate proposed for speaker diarization is diarization error rate (DER) (Anguera et al., 2012). This error rate is a combination of three types of errors made by a speaker diarization system. The errors include:

- false alarm errors – segments that are falsely detected as speech segments,
- miss errors – segments that contain speech from one of the speakers, but are missed (flagged non-speech) by the diarization system,

- speaker error – clustering error, which incorrectly labels segments,

The last type of error is the most difficult to verify, since the diarization system has no prior knowledge of speakers and therefore assigns its own labels to segments. Hence, the labeling produced by the diarization system does not correspond to labels from ground-truth. The figure below shows an example problem. It shows all three types of error in a diarization system (false-alarm, miss, and incorrect labeling of speakers). Figure 5.5 highlights the types of error that are made during diarization.

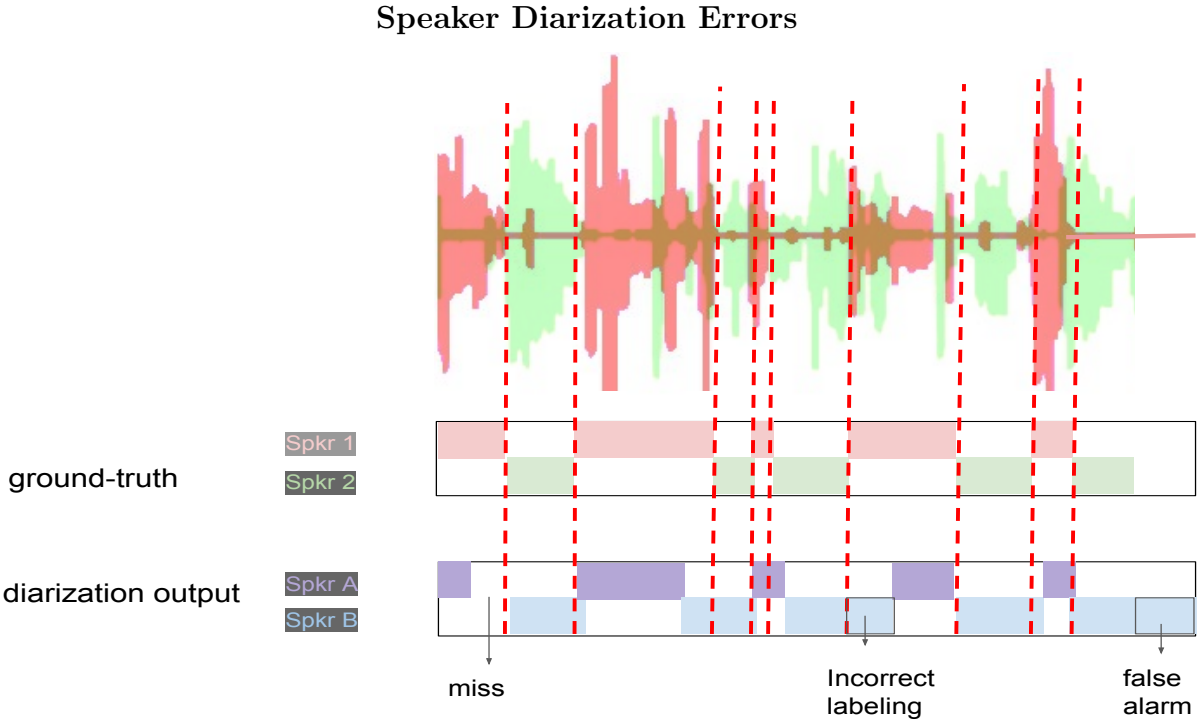


Figure 5.5. Shows three types of errors made by a diarization system. 1) false alarms: segment that does not contain speech is labeled by diarization system. 2) miss: segment containing speech is not labeled by diarization system. 3) incorrect labeling: the third type of error assumes that A is 1 and B is 2. It is up to the diarization error rate calculator to make this assignment.

It is also shown that the labels assigned by the speaker diarization system $(A, B)$ are different from ground-truth labels $(1, 2)$. It is up to the diarization error rate calculator to recognize the correct label correspondence, $A : 1$ and $B : 2$. This is done by considering all

possibilities, in this case $(A:1, B:2)$ and $(A:2, B:1)$, and returning the assignment that results in minimum labeling error.

The formulation proposed by NIST for diarization error rate is (Anguera et al., 2012):[4]

$$DER = E_{spkr} + E_{miss} + E_{fa}, \tag{5.13}$$

where:

$$E_{spkr} = \frac{\sum_{s=1}^{S} dur(s)(min(N_{groundtruth}(s), N_{diarization}) - N_{correct}(s))}{T_{score}} \tag{5.14}$$

$$E_{fa} = \frac{\sum_{s=1}^{S} dur(s)(N_{diarization}(s) - N_{groundtruth}(s))}{T_{score}} \tag{5.15}$$

$$E_{miss} = \frac{\sum_{s=1}^{S} dur(s)(N_{groundtruth}(s) - N_{diarization}(s))}{T_{score}} \tag{5.16}$$

in which $S$ is the total number of segments. The variable $dur(s)$ is the duration of segment $s$. $N_{groundtruth}(s)$ is the number of speakers in segment $s$ provided by the ground-truth. $N_{diarization}(s)$ is the number of speakers in segment $s$ hypothesized by the diarization system. $N_{correct}(s)$ is the number of speakers correctly matched by the diarization system. Finally, $T_{score}$ is the total scoring time.

CRSS-SpkrDiar uses an evaluation Perl script provided by NIST that calculates DER alongside the individual errors described above. This script requires a certain input format for the labels, called rich transcription time marks (RTTM). CRSS-SpkrDiar contains labelToRTTM and glpkToRTTM binaries that convert Kaldi vectors to RTTM format. The output of the GLPK clustering module can also be converted into RTTM using glpkToRTTM.

### 5.6.2 Comparing distance measures

This section briefly compares CRSS-SpkrDiar performance under the various distance measures described in Sect. 5.4 and 5.5. Experiments are conducted on AMI meeting data. Each meeting recording, referred to as a *session*, contains a minimum of 4 speakers. Other than the session audio file, no additional information is provided to the diarization system. However, as mentioned before some of the distance metrics, PLDA and conditional Bayers, require external data (i.e., development data). Development data is provided in the form of i-Vectors from NIST SRE challenges. For the purposes of these experiments i-Vectors were extracted for over 600 NIST SRE04,05,06 speakers. Multiple i-Vectors from different record-

[4]Unfortunately, NIST has removed the online evaluation plan from its website. The equations provided here are from Xavier Anguera's PhD thesis (Miro, 2007).

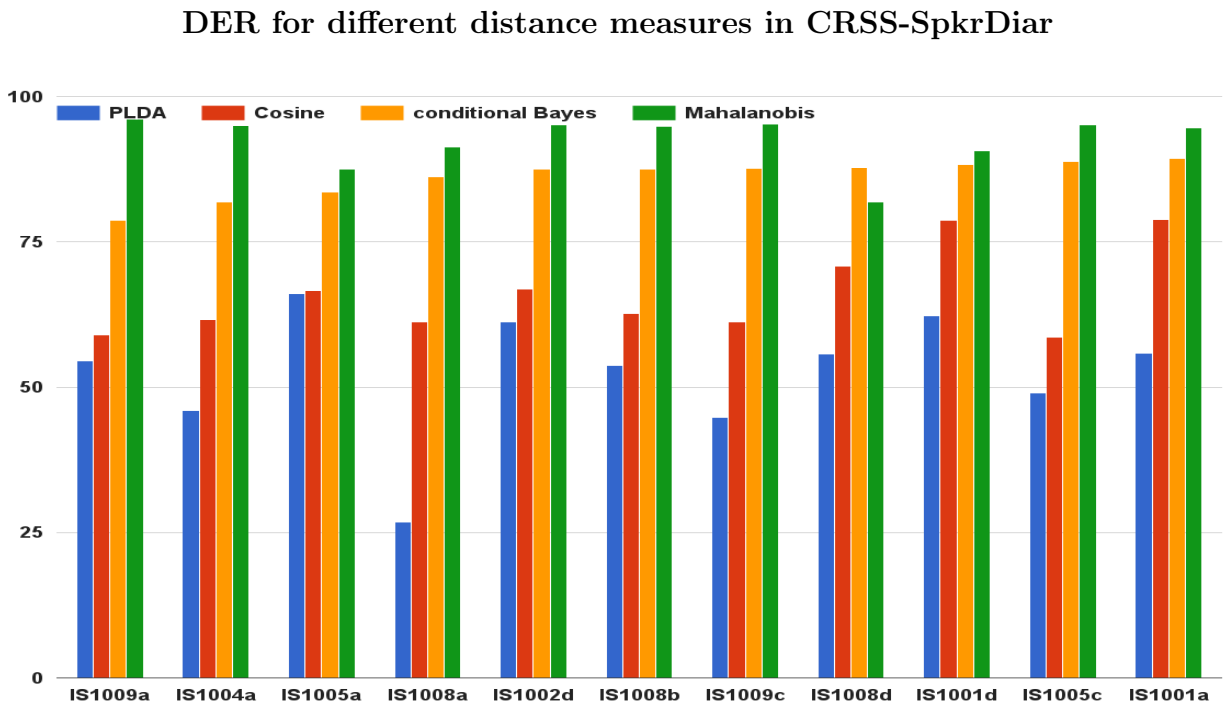


Figure 5.6. Comparison of diarization error rates calculated for different distance measures implemented in CRSS-SpkrDiar. DER is calculated for 11 AMI meetings (IS1009a-IS1001). Although absolute DER values have considerable room for improvement, it is clear that PLDA scoring significantly outperforms other distance measures.

ings were used for each speaker to provide sufficient channel variability. The PLDA model used here does not perform dimension reduction on the original 100-dimensional i-Vectors.

Readers are advised not to examine Fig. 5.6 based on absolute DER values. The purpose of this figure is to compare the performance of different distance measures presented in this chapter. The novel approach of using PLDA-based scores in ILP clustering significantly outperforms other distance measures. Figure 5.6 also shows that conditional Bayes only slightly outperforms Mahalanobis distance. This observation is reasonable, since Mahalanobis and conditional Bayes essentially follow the same form and the difference is in the choice of covariance matrix.

## 5.7 Future Work

As mentioned throughout this chapter, CRSS-SpkrDiar is a promising new research platform for speaker diarization. Every few weeks, we, the developers, receive requests for collaboration and questions about the state of the project from researchers around the globe. This chapter provided an outline of CRSS-SpkrDiar modules implemented so far. But as mentioned before, CRSS-SpkrDiar is a work in progress, both as a research platform and a means of achieving state-of-the-art speaker diarization performance. It was pointed out that speech activity detection and overlap detection modules are integrated into CRSS-SpkrDiar. A major improvement to CRSS-SpkrDiar would be to re-implement these components using Kaldi APIs, instead of using existing code in a plug-and-play manner. Finally, in addition to the two main components of speaker diarization, segmentation and clustering, some minor additions can help significantly improve performance. These additions include: alternative clustering algorithms, Viterbi resegmentation, and GMM-based speaker modeling. Considering the current state of CRSS-SpkrDiar and the reported error rates, provided sufficient time and effort, state-of-the-art diarization performance is well within our reach.

## 5.8 Summary

Chapter 5 introduced CRSS-SpkrDiar, a speaker diarization system. The objective of CRSS-SpkrDiar is to create an all-encompassing research platform that supports state-of-the-art speaker and speech recognition, in addition to its primary objective, which is speaker diarization. This chapter focused on to main components of a speaker diarization system: 1) segmentation and 2) clustering. CRSS-SpkrDiar is considered a work in progress and further studies on this tool-kit are in development.

# CHAPTER 6

# APPLICATIONS[1]

Previous chapters focused more on various technical aspects of co-channel speech and less on integrating the proposed techniques with other applications. This chapter highlights collaborative studies that address co-channel speech analysis for two problems: word-count estimation and in-vehicle speech pertaining to driver safety systems.[2] I consider these studies a driving force during my work as a PhD student, since research without outside stimulus can sometimes be mundane and out of touch. Collaborative studies, on the other hand, open up room for discussion and often lead to constructive feedback. Each of the studies presented in this chapter have led to major advancements in this dissertation.

The contribution of this chapter is:

- use overlap detection to improve word-count estimation in real environment noise conditions.
- use overlap detection to predict competitiveness in conversations in in-vehicle speech analysis.

Prior to the study on word-count-estimation, the main overlap detection algorithm developed for this dissertation was Gammatone sub-band frequency modulation (GSFM; see Sect. 2.5

[1]Portions of this chapter were adopted from published material with the authors' full consent. Shokouhi, Navid, Amardeep Sathyanarayana, Seyed Omid Sadjadi, and John H. L. Hansen, "Overlapped-speech detection with applications to driver assessment for in-vehicle active safety systems," In Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP) ©2013 IEEE.

Shokouhi, Navid, Ali Ziaei, Abhijeet Sangwan, and John H. L. Hansen, "Robust overlapped speech detection and its application in word-count estimation for Prof-Life-Log data," In Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP) ©2015 IEEE.

[2]Word-count estimation in co-channel speech is a collaboration between myself, Ali Ziaei, Abhijheet Sangwan, and my advisor Prof. John Hansen. At the time of the study, Ali Ziaei was a fellow PhD student at CRSS, Abhijeet Sangwan was (and currently still remains) a research professor in the department of Electrical Engineering at UT-Dallas.

In-vehicle speech analysis studies are a collaboration with Amardeep Sathyanarayanan, Omid Sadjadi, and Prof. John Hansen. Amardeep and Omid were both PhD students at CRSS during the time of these studies.

in Chapter 2). As we have learned, GSFM is highly susceptible to noisy conditions. This observation was made while evaluating overlapped speech detection in Prof-life-log data, a dataset with various types and amounts of noise (Ziaei et al., 2013). Pyknograms were the solution to a long-lasting effort of developing a robust overlap detection method for noisy conditions (see Sect. 2.4 and 2.6 in Chapter 2).

In-vehicle speech analysis contributed to a significant novelty in this dissertation's perspective in addressing co-channel speech (in terms of distinguishing co-channel speech from overlap). Prior to the collaborative study on in-vehicle speech, the sole focus of this dissertation was on examining overlapped speech. It was during this project that the importance of other conversational traits, such as turn-takings and speaker response time, became more visible and prompted further investigation.

The remainder of this chapter provides a brief description of word-count estimation, Sect. 6.1, and in-vehicle driver safety systems, Sect. 6.2.

## 6.1 Word-Count Estimation in Co-channel Speech

The ability to estimate the number of words spoken by an individual over a certain period of time is valuable in second language acquisition, health-care, and assessing language development. Word-count values are also useful in the analysis of massive audio data, such as the Prof-life-log corpus (Ziaei et al., 2013). Prof-life-log is a speech corpus that contains long durations of audio recordings. In this collection, the primary speaker wears a portable LENA recording device (Dongxin et al., 2008) throughout a work day. Although the primary speaker is always the same individual, he frequently interacts with different people. An estimate of the primary speaker's speech activity (i.e. word-count) facilitates analyses that predict the level of productivity and helps determine areas in the recording that are more "valuable" for further processing. The proposed word-count estimator in this study is tested on long duration files from the Prof-Life-Log corpus.

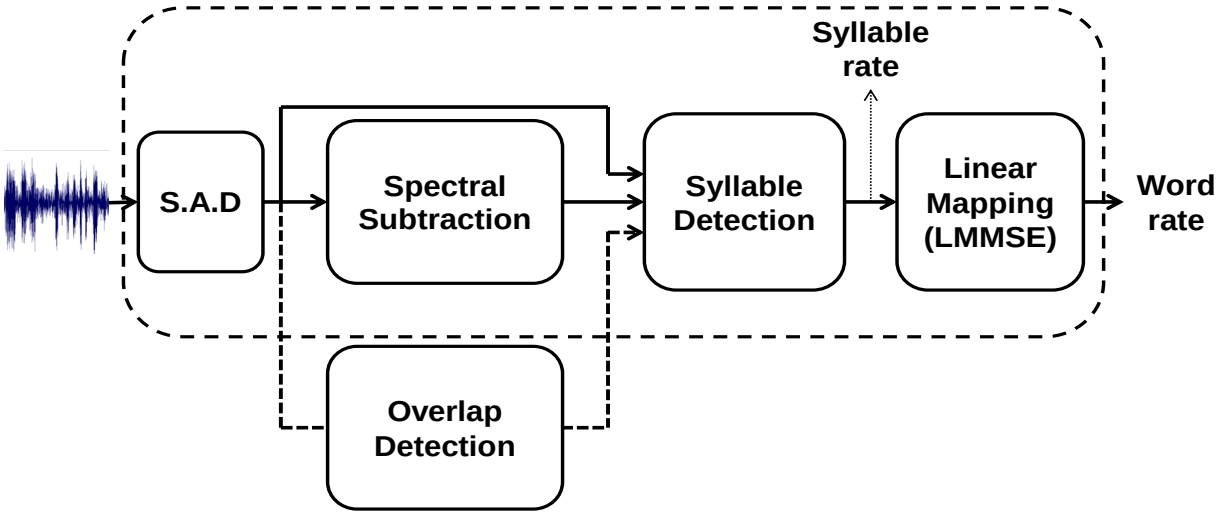


Figure 6.1. Word-count estimation system configuration. The overlap detection system is shown as an addition to the original system.

Establishing a robust automatic framework to achieve high accuracy is non-trivial in realistic scenarios due to various factors such as different conversation styles or types of noise that appear in audio recordings, especially in multi-party conversations (i.e., co-channel speech). This section proposes to use overlap detection to estimate the likelihood of overlapping speech in a given audio file in the presence of environment noise. The resulting overlap information is embedded into a word-count estimator, which uses a linear minimum mean square estimator (LMMSE) to predict the number of words from syllable rates, which are easier to detect using only acoustic information. Figure 6.1 shows an overview of the proposed word-count estimator. The overlap detection system acts as an augmented module in the system, similarly to how speech activity detection removes silence for the word-count estimation (WCE) system.

### 6.1.1 Word-count estimation

The word-count estimator is adopted from previous studies (Ziaei et al., 2014), which estimate the number of words per unit time by applying a linear transformation to the syllable

rate. Syllable rates are calculated based on a modified version of the *mrate* algorithm (Wang and Narayanan, 2007) using acoustic characteristics of the signal: pitch, smoothed spectrogram. This algorithm detects the location of syllables in a given speech segment. The number of detected syllables per unit time are used to calculate syllable rates throughout the audio. As (Ziaei et al., 2014) show, the help of alinear minimum mean square estimator (LMMSE) can generate linear coefficients that map syllable rates to the number of words per unit time, as shown below:

$$\tilde{\mathbf{a}} = argmin_{\mathbf{a}}\Big\{\frac{1}{N}\sum_{\mathbf{a}}\big(W_r(n) - \mathbf{a}S_r(n)\big)^2\Big\}, \tag{6.1}$$

where $W_r(.)$ and $S_r(.)$ are the word-count and syllable rates at any given time, respectively. $n$ indicates the index of a given time segment and $N$ is the total number of segments used to train the linear transformation parameter, $\tilde{\mathbf{a}}$. The transformation parameter, $\tilde{\mathbf{a}}$, is a vector comprised of a bias factor and a linear coefficient. In cases where the bias factor is non-zero, $S_r(n)$ is replaced by $\big[S_r(n)\ 1\big]^T$. The linear transformation parameter(s) can be trained using manually transcribed background conversational data. For this study, we rely on a subset of Prof-Life-Log data that has been transcribed for word-counts.

In (Ziaei et al., 2014), higher accuracy is obtained by introducing speech activity detection (SAD) (Sadjadi and Hansen, 2013) and spectral subtraction to the front-end of the WCE. SAD reduces false-alarms by omitting non-speech regions, a decrease which helps avoid detection errors made by the syllable detector. Spectral subtraction enhances speech regions (Boll, 1979), allowing the syllable detector to identify voiced regions more accurately. None of these techniques, however, are able to address the issue of overlapped speech. Therefore, the novelty in this study is to use overlap detection as an additional layer of data pruning to reduce false alarms.

An estimation of the location and amount of overlapped speech in a given speech segment can be combined with SAD labels to supply an additional layer of data pruning before syllable

detection. Overlap detection is fed to the syllable detector as additional information (see Fig. 6.1). First, SAD is performed on the raw data to detect speech locations. From SAD labels, non-speech regions are used to estimate the noise level in each short segment and submitted to the spectral subtraction algorithm. Speech-only segments are passed to the syllable detector after applying spectral subtraction. Finally, syllable rates (calculated by dividing the number of syllables by the segment length) are transformed into word-count rates using LMMSE coefficients. In our proposed system, overlap detection outputs are combined with SAD results, to provide an extra layer of data pruning.

The overlap detection algorithm used in this framework is based on Pyknograms, which are described in Chapter 2. Pyknograms were chosen here as a robust solution to the highly noisy data in Prof-life-log. In Pyknograms, non-resonant speech is emphasized over background noise and unvoiced speech. This allows improved performance in noisy conditions, while simultaneously improving syllable detection, due to the high correlation between voiced speech and syllables.

Word rates are extracted from 5 days of prof-life-log recordings. Each day contains roughly 6 to 8 hours of audio, labeled to include the transcriptions of the primary speaker's speech, speaker labels (primary vs. secondary), and the type of environment in which the recordings take place. We have mostly concentrated on environments that are more likely to contain overlapped speech, such as multi-party meetings and conferences. The sampling frequency from the LENA device is $44.1kHz$, which we have down-sampled to $8kHz$. Table 6.1 shows over 35% improvement in relative mean square error after removing overlapped regions.

Table 6.1. WCE performance in Prof-Life-Log with respect to overlapped speech.©2015 IEEE

| $\frac{minimum\ mean\ square\ Error}{\#ofwords}$ | |
|---|---|
| overlaps NOT removed | 5.71% |
| overlaps removed | **3.69%** |

## 6.2 In-vehicle Conversation Analysis

Many in-vehicle conversations are beneficial in keeping drivers alert and active, however there are also instances where a competitive conversation may adversely influence driving performance. Identifying such scenarios can improve vehicle safety systems by fusing the knowledge obtained from conversational speech analysis and vehicle dynamic signals. This section briefly describes how smart portable devices can be incorporated to create a unified platform and record in-vehicle speech as well as vehicle dynamic signals required to evaluate driving performance. This study shows that turn-takings and overlapped speech segments, as conversational speech cues, under certain conditions deviate from normal driving patterns.

Auditory based distraction caused by in-vehicle conversations is difficult to investigate. The main issue being that not all speech activity is considered distractive to drivers. The study in (Sathyanarayana et al., 2012b) used speech activity detection to track the impact of any speech activity on driving performance. The conclusion being that although some impact is observed, driving performance is highly affected by the type of conversation. In this study, we further investigate the possible consequences of drivers' involvement in *competitive* conversations while driving. It is expected that active engagement conversations results in deviations from usual driving patterns, which are identified through the analysis of different driving maneuvers.

### 6.2.1 System Description

Performing certain secondary tasks while driving is likely to adherently impact driving performance. Some tasks are inevitable or difficult to restrict, such as controlling the navigation system. However, other secondary tasks such as engaging in a conversation with other passengers or cell-phone conversations can be controlled if they are determined to compromise driving performance. This sections shows a feedback system which not only evaluates variations in driving performance but also helps mitigate the influence of in-vehicle speech if it

is found to adversely influencing the driving performance (see Figure 6.2). There are two separate subsystems in the proposed framework. The first subsystem evaluates the driving performance by identifying maneuvers and then comparing them with the driver's regular driving patterns. If a maneuver is recognized as "abnormal", the driver's in-vehicle speech involvement is monitored to see whether that is the cause for unusual driving behavior.

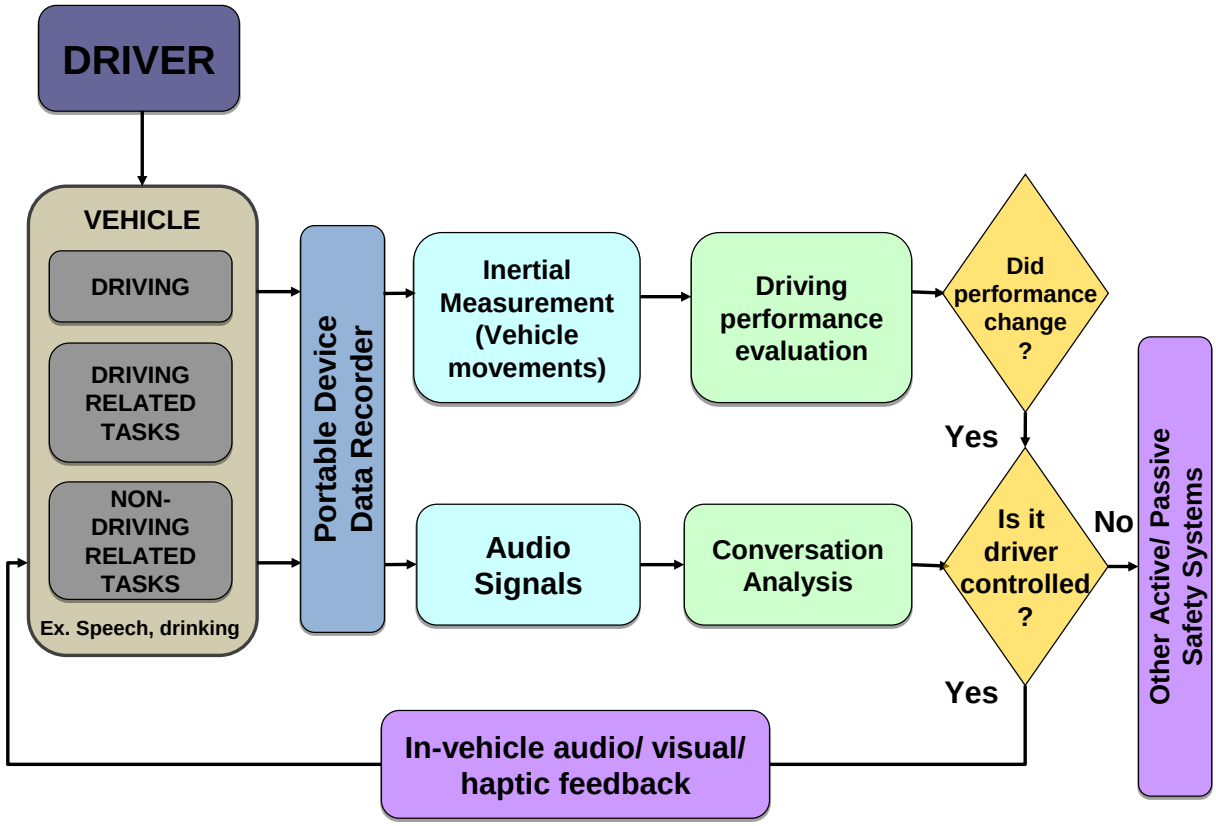


Figure 6.2. System description

The second subsystem evaluates the driver's involvement in conversations. An addition to previous studies on in-vehicle speech (Sathyanarayana et al., 2012b) is that rather than considering any in-vehicle speech activity, the emphasis here is on the driver's involvement in conversations. This involvement is analyzed by measuring the amount of overlapped speech segments as well as turn-taking rates during conversations. One of our objectives was to demonstrate that these two metrics are reliable indicators of the driver's focus on speaking

as opposed to the primary task, which is driving. If in-vehicle speech activity is identified to adversely impact the driver's performance, it can be controlled passively by providing an initial warning feedback to the driver and/or co-passengers to halt the conversation until the vehicle state returns back to normal. Monitoring driving performance continues and any secondary tasks performed by the driver, other than those essential to driving, could be sequentially cut off. The uniqueness of this system is the ability to identify driving performance variations and isolate the source of distraction.

The technical details of the driving evaluation system falls beyond the scope of this dissertation. For the curious reader, it suffices to know that performance is measured by statistically modeling driving maneuvers using vehicle dynamic signals (e.g., steering wheel angle, speed, acceleration) (Sathyanarayana et al., 2013). Maneuvers form the basic building blocks of driving routes, hence analyzing them can be employed as a key component in understanding driving performance. Variations in driving performance can be recorded by observing how each maneuver is executed and comparing the characteristics of maneuvers that fall under the same category.

### 6.2.2 Conversation Analysis

As mentioned in Section 6.2.1, the purpose of utilizing audio data is to analyze the conversation taken place in the vehicle and determine whether it is causing driver distraction. We speculate that the level of competitiveness in a conversation should correlate with driving performance. In order to measure competitiveness in a conversation, two features are utilized. The first is the turn-taking rate. An increase in the number of turn takings per time unit can imply that speakers are taking interest in the conversation. Additionally, the amount of overlapped speech is considered a potential competitiveness feature (Schegloff, 2002).

By now the readers are likely familiar with overlap detection algorithms, but turn-taking has not been covered in this thesis. Although there are more sophisticated ways of estimating turn-takings rates in a conversation (using speaker diarization). This study uses speech activity detection (SAD) to detect start and end-points of active speech regions in a conversation. The start-points are labeled as the triggering points of a turn in the conversation. Since this definition for turn taking may not always imply that both speakers are involved in a conversation, for example in instances when one of the speakers pauses in between sentences, the amount of overlapped speech in regions with a high turn taking rate is used to assure that both speakers are involved.

## 6.3 Data Description

Since this study focuses on understanding the influence of in-vehicle speech on the driver, care should be taken to minimize influence from other modalities on the driver. An experimental setup was conducted that allowed drivers operate the UTDrive vehicle under real-traffic conditions (Angkititrakul et al., 2007). The data was collected under similar weather and traffic conditions for all drivers and the route consisted of residential areas and highways and took place in an average of twenty minutes per session. In the past few years, the research community has shown an increasing interest in using portable devices (sensor loaded smartphones and tablets) to instrument a vehicle and use it as a pseudo-data collection platform. It has also been shown that using sensor data from these portable devices yields to comparable results in maneuver recognition CAN-bus signals obtained from instrumented vehicles (Sathyanarayana et al., 2012a, 2013). In this study, data was collected in the UTDrive instrumented vehicle (UTDrive) along with the portable device mounted in the car (Sathyanarayana et al., 2012a). However, only the sensor information from the portable device has been used for the entire analysis. Using a 10.1" Samsung Galaxy Tablet and an

Android OS as the portable device platform, an Android application has been developed to collect all the available sensor information on the device synchronously. The available sensors and derived information include a camera, microphone, accelerometer, gyroscope, magnetometer, orientation, compass and GPS signals. A detailed description of the UT-Drive App for portable device and the sensor information can be found in(Sathyanarayana et al., 2012a).

Data were designed with the intention to create competitive behavior in the driver. Drivers were asked to perform tasks, which activate their competitiveness and involvement in order to increase the amount of overlapped speech (Schegloff, 2002) and turn-takings. However, the fact that the purpose of the study was to collect competitive conversations was hidden from the drivers to avoid self-consciousness. The driving route was divided into four segments, repeated in two phases. In the first phase, the driver drives through the complete route without performing any secondary task to become familiar with the route and the vehicle. In the second phase the driver is asked to perform a different task in each segment of the route. Four different tasks are chosen to include as many variations of conversational speech as possible. The tasks are described below:

- Segment 1: At the beginning, in order to break the ice the passengers will initiate a simple conversation by asking the driver questions about casual topics such as the weather (Other topics may be chosen).

- Segment 2: In this segment one of the passengers picks an object and the driver and the other passenger are supposed to guess the object using the hints provided to them. Whoever guesses first is the winner. This game is chosen to increase turn-taking and overlapping speech segments.

- Segment 3: A set of TIMIT sentences are played through a portable tablet and the driver is required to repeat each sentence before the next sentence is played.

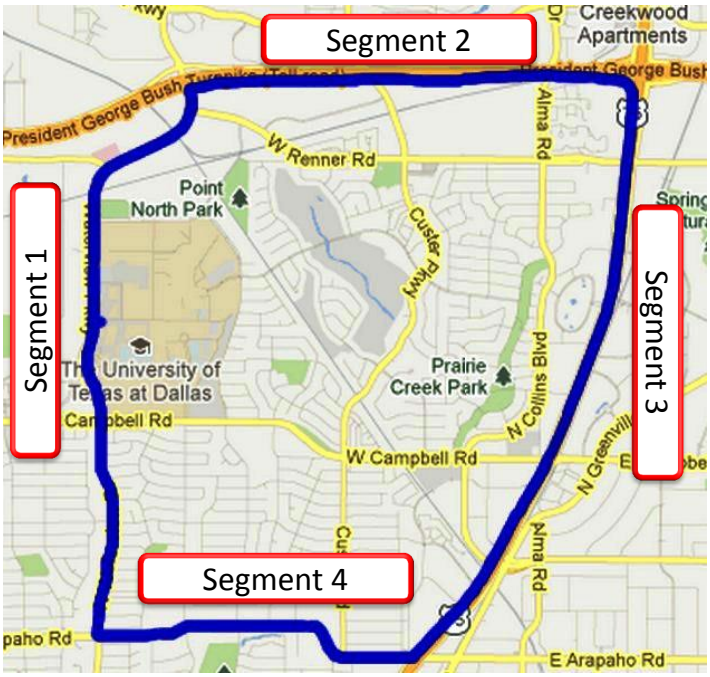


Figure 6.3. Driving route with different conversational task segments.

- Segment 4: An argument is initiated by one of the passengers. This second conversation is more involved compared to that in segment 1. The difference here is that the drivers opinion on a debatable topic is asked and based on his/her answer, the passengers take the other side and try to argue on the subject. Each of the tasks described above takes place on one of the legs in the route as labeled on the map in Figure 6.3.

### 6.3.1 Experimental results

An advantage of this experimental setup over previous studies (Sathyanarayana et al., 2012a, 2013, 2012b) is that both the in-vehicle speech data and the data required for maneuver recognition are recorded by the portable device. This results in a more concise and cost effective data acquisition platform.

Statistical information from the inertial measurement sensors such as accelerometer and gyroscope of the portable device were extracted on a per frame basis (once per second).

Statistical information such as maximum lateral acceleration, mean of the vehicle speed, variance of yaw-gyroscope (refer to (Sathyanarayana et al., 2012a) for a detailed list) form the dominant feature space used in training the maneuver specific models. Using Support Vector Machines (SVM), the maneuver segments are recognized and classified with a high average accuracy of over 90%. Once classified, driving performance is evaluated for each maneuver according to its class. Thresholds are appropriately set in the feature space to identify any abnormal or risky driving patters. Examples of the driving performance evaluation is shown in Figure 6.4. This figure compares a drivers performance on the same route with and without the stimulus of competitive conversations. The regions marked green are locations where the driver drove similar to his usual driving pattern. Yellow regions correspond to where the driver showed slight variations in the driving pattern. The red regions are locations where the driver executed an abnormal or risky maneuver. All regions concur with the visual and perceptual verification obtained by looking at the video from the data recordings.

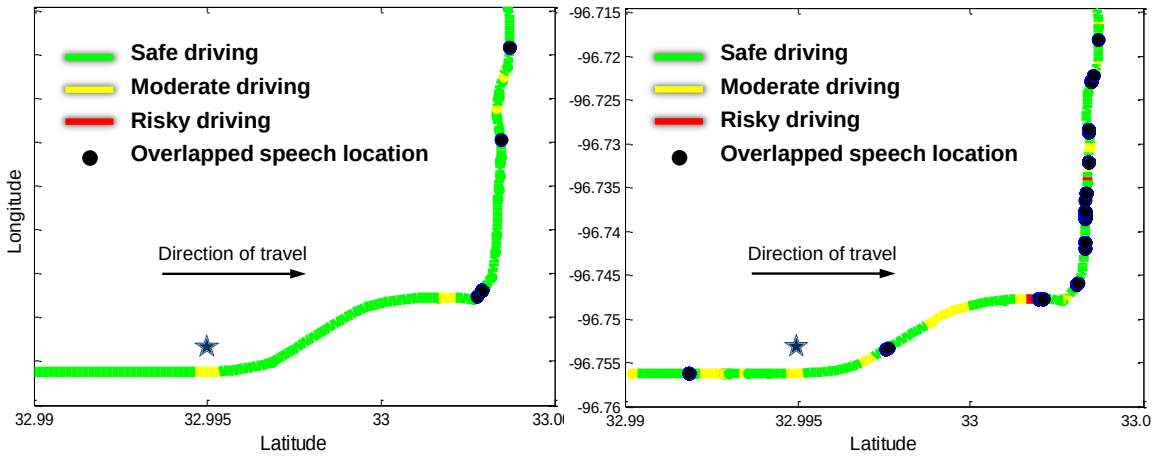


Figure 6.4. Driving performance evaluation on a section of the route in two phases. Left (phase 1)- performance with minimal conversations. Right (phase 2)- performance drop as a result of the increase in the amount of overlapping speech.©2013 IEEE

As Figure 6.4 suggests, there are some instances in the route where no direct relationship is observed between overlapped speech and performance-drop, see the area marked by a star in the figure. This was expected, since there are always other sources that can cause

irregularities in the driver's maneuver execution patterns. Hence, speech related features should be analyzed in more detail to confirm that the conversation is the source of distraction. With this intention, the patterns of turn-takings and overlapped speech segments were jointly investigated over time. Turn-taking and overlapped speech rates are defined as below:

$$ovl_{rate} = \frac{\text{Numberofoverlappedsamplesinwindow}}{\text{windowlength}}$$

$$tt_{rate} = \frac{\text{Numberofturntakingsinwindow}}{\text{windowlength}}$$

Figure 6.5 depicts the average turn-taking and overlapped speech rate in conversations before observing significant drop in driving performance. Each plot belongs to a different scenario.

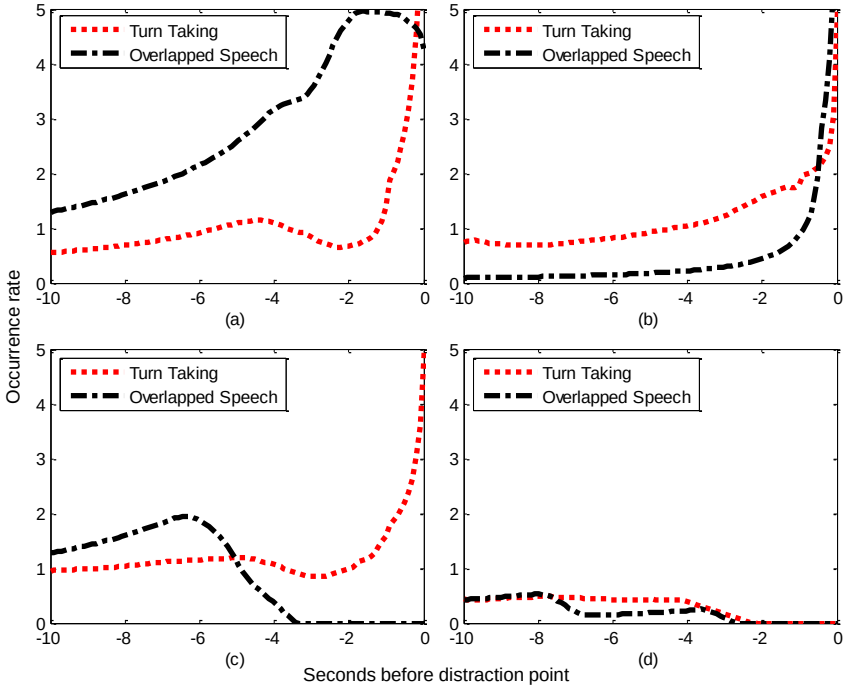


Figure 6.5. Turn-taking and overlapping speech rate before observing the first major drop in performance in different scenarios.

If both the turn-taking and overlapped speech rates increase before the performance drops, the conversation can be a potential source of distraction. Figures 6.5-a and b are more interesting, since both features increase with time. Figure 6.5-c was also observed in our experiments, where the overlapped speech rate increases and goes back to 0 per second before driving performance drops. Figure 6.5-d shows an instance where turn-taking and overlapped speech rate significantly drop prior to any changes are observed in driving performance, lowering the possibility that the conversation being the source of distraction.

## 6.4 Summary

The final chapter highlights a selection of collaborative studies between co-channel speech analysis and other signal processing applications. The content was presented in the form of an applications chapter. The first section describes how overlap detection can be integrated into a word-count estimation system to reduce false-alarm errors in detecting syllables. The second section uses co-channel speech analysis, namely turn-taking rate estimation and overlap detection, to measure speakers' attentiveness in conversations. The predicted "attentive" is used in a driver safety system to identify in-vehicle conversations that jeopardize driver and passenger safety.

# CHAPTER 7

# CONCLUSIONS

This study presented a detailed description of co-channel speech analysis in the context of speaker recognition. As repeatedly mentioned throughout the course of this thesis, co-channel speech refers to single-channel audio signals with more than one speaker. Great care was taken to highlight the difference between overlap and co-channel speech. While co-channel refers to any signal with more than one speaker, overlap refers to parts of co-channel audio where more than one speaker is speaking. The distinction between overlap and co-channel is the first contribution of this thesis. Although this contribution may appear trivial at first glance, it was shown that the majority of co-channel research considers overlap synonymous to co-channel speech, a misconception that reduces the applicability of some attempts to compensate overlapped speech in speaker recognition. In other words, addressing overlap in co-channel speech is not sufficient for a large class of speaker recognition problems. This narrative is carried out throughout the first few investigations of this study. The following sections are some of the major contributions of this thesis.

## 7.1 Overlapped speech analysis in speaker recognition

This study considers the impact of overlapped speech on speaker recognition performance. Speaker recognition is typically manifested through speaker verification experiments. It was shown that replacing single-speaker audio data with overlapped data reduces verification performance to up to one order of magnitude. For example, equal error rates (EER) increase from $1-2\%$ for single-speaker audio from the GRID corpus to over $20\%$ for overlapped data (depending on the nominal signal-to-interference ratio). This observation shows the legitimacy of concerns in past studies, which prioritize overlap over other forms of co-channel data. The traditional approach to deal with such drastic performance drop is to measure/detect

overlapped segments in speaker recognition experiments. Therefore, the second stage of this study provides an extensive investigation of overlap detection methods.

## 7.2 Overlap detection

Overlap detection provides a unique perspective in highlighting the differences between single-speaker and overlapped speech. The algorithms proposed in this study focus on the harmonic structure of speech. Speech harmonics have traditionally been used as a way to identify overlapped speech. The fact that speaker recognition is highly influenced by voiced speech further motivates this approach. Harmonics are an important component of voiced speech and therefore, harmonic based analyses of overlapped speech fits well with the theme of this study (i.e., speaker recognition).

The two methods proposed for overlap detection are: 1) Pyknograms, and 2) Gammatone sub-band frequency modulation (GSFM). Pyknogram extraction is a 2 step process of obtaining a binary mask for time-frequency units corresponding to the amplitude spectrogram. Frequencies across the spectrogram are first estimated using the Teager Energy Operator (step 1). The estimated frequencies are then pruned to only include prominent resonances (step 2). Pyknograms provide two important features that are useful for overlap detection: 1) unlike many existing speech representations, Pyknograms do not depend on the number of speakers; 2) Pyknograms are effective in suppressing non-harmonic speech. This suppression provides robust performance in noisy conditions. In addition to Pyknograms, GSFM was also proposed as a way to magnify the presence of multiple harmonics in time-frequency units. GSFM incorporates the non-linearity of sinusoidal frequency modulation to obtain frequency modulation spectra. The relative roll-off of FM spectra is then used to summarize the information in each time-frequency unit. Evaluations show that although in controlled conditions GSFM outperforms Pyknograms for overlap detection, while Pyknograms are significantly more reliable under noisy conditions.

## 7.3 Incorporating overlap detection in speaker recognition

In addition to introducing overlap detection methods, a novel technique is proposed that uses overlap detection decisions as quality measures for speaker verification experiments. Using overlap detection as quality measures (aka meta-data) is more desirable compared to the traditional approach, which is to detect and remove overlapped segments. The advantage of quality measures is in the fact that not all overlapped speech should be thrown away, especially when the amount of available data is limited. The algorithm used to fuse quality measures with speaker recognition decisions is called Q-stack, which concatenates multiple scores and summarizes the final speaker verification decision using support vector machine (SVM) certainties. Fusing overlap meta-data with speaker verification scores relatively reduces speaker verification error rates by approximately 20%.

## 7.4 Modified PLDA for speaker recognition in co-channel speech

Although overlaps can significantly impact speaker recognition performance, the practicality of this concern is debatable. Therefore, this study also focused on evaluating speaker recognition performance in the more general case of co-channel speech, rather than overlap. It was shown that although overlap is damaging to speaker recognition performance, the impact of co-channel is much more significant. In the case of Switchboard2 telephone conversations, the impact due to non-overlapping co-channel speech is observed as an 18% increase in EER (single-speaker performance is 5%). Meanwhile, the increase in EER due to overlapped speech is slightly over 2%. This puts the impact of overlap vs. co-channel speech in perspective when addressing real conversational speech. In an effort to compensate co-channel in realistic speaker recognition problems, a standard i-Vector/PLDA system was evaluated. Many approaches were investigated to improve probabilistic linear discriminant analysis (PLDA) for co-channel speech.

The first method was to add co-channel data to PLDA training (mixed-PLDA). Mixed-PLDA was presented to treat speaker interference in the same manner PLDA addresses channel mismatch. Although mixed-PLDA was proposed in this study, it is also considered a baseline; since comparing proposed systems with standard PLDA, which is not designed to compensate for co-channel data, is not a fair assessment. A second method proposed in this study is dual-eigenvoice PLDA (dePLDA). dePLDA uses two identical eigenvoice matrices to model within- and between-speaker variability. The difference between dePLDA and standard PLDA is in replacing the eigen-channel matrix with a second eigenvoice. The third method is co-channel aware PLDA (caPLDA), which proposed alternative formulations to PLDA for speaker recognition in co-channel speech. These alternative formulations replace within-speaker variability with a linear combination of between- and within-speaker covariances. Replacing the traditional within-speaker covariance is motivated by the fact that secondary speaker interference poses a more significant impact on i-Vector distributions, compared to channel variability. Different coefficients are investigated in the proposed linear combination framework. Results show that by a careful consideration of linear combination parameters, speaker recognition performance is expected to improve using caPLDA. dePLDA also provides significant improvement.

## 7.5 CRSS-SpkrDiar

A speaker diarization research platform called CRSS-SpkrDiar was presented in this study. Speaker diarization addresses the problem of "who spoke when?", which is a relevant question for co-channel speech signals. Therefore, diarization is considered an important aspect of speaker recognition in co-channel speech. In a sense, while most of this thesis considered speaker recognition over entire co-channel audio-streams, speaker diarization addresses speaker recognition within a co-channel signal. CRSS-SpkrDiar is a C++ library that includes the implementation of state-of-the-art speaker diarization techniques. This study

presents the main components of CRSS-SpkrDiar while providing sufficient details related to speaker diarization in general. CRSS-SpkrDiar is considered a major contribution of this study on co-channel speech and is presented as a stepping stone for future work.

## 7.6 Applications

As a final stage for this study, two applications of co-channel analysis in other signal processing applications were investigated. It was shown that the techniques developed for overlap detection and co-channel analysis in general could be used to improve word-count estimation in realistic conversational co-channel data. Co-channel estimation was also proven useful in assessing conversations in in-vehicle conversations, in an effort to monitor the impact of driver conversations with driving performance.

## 7.7 Future work

A wide range of topics in co-channel speech were covered throughout this study. Care was taken to assure that everything that affects speaker recognition be fully analyzed and some novel techniques were proposed.

The algorithms used for overlap detection mostly consider unsupervised signal processing techniques. A natural improvement on unsupervised methods has always been to investigate these methods in supervised frameworks. A more elaborate setup to track harmonic trajectories in Pyknograms involves using a hidden Markov model (HMM) to model Pyknogram patterns. This allows for a supervised classification of speech segments into overlap and single-speaker speech. The proposed system could use an initial segmentation of a given audio stream based on the Bayesian Information Criterion (BIC segmentation). The shorter segments are then compared against two pre-trained HMMs, one for overlapped and the other for single-speaker speech. The initial BIC segmentation is to allow for detection

on shorter segments, since overlapped speech is considerably less frequent in a conversation compared to the total amount of speech. The combination of BIC segmentation and HMM-based classification allows for a practical overlap detection mechanism that fits well with analyzing conversations recorded in real conditions (as opposed to artificial datasets with simulated overlaps). Among other benefits of this framework is its compatibility with speaker diarization tools in which overlapped speech is considered a nuisance.

Although the developments in modified PLDA algorithms resulted in some improvement for co-channel PLDA, further efforts can find an optimal structure to model within-speaker variability for co-channel speech. The assumption is that for co-channel audio, each class correponds to a fixed primary speaker and varying secondary speakers. The solution proposed here was to share information from inter-speaker variabilities to improve performance. An alternative model could use a separate dataset to obtain inter-speaker information, therefore removing the convergence issues associated with sharing parameters in the PLDA training process. On that note, EM convergence could be an interesting starting point for analyses in modified PLDA, especially in the case of dual-eigenvoice PLDA.

The third component of this dissertation that has the most potential to contribute to future studies is CRSS-SpkrDiar. This tool-kit was designed with the intention of providing comprehensive source code that directly relates to problems addressed in this study (co-channel, overlap, speaker recognition, speaker diarization). CRSS-SpkrDiar provides full control to users and is constantly under development to alleviate future studies.

# APPENDIX: I-VECTOR/PLDA SPEAKER RECOGNITION

This appendix is presented for the curious reader who is interested in becoming more familiar with the i-Vector/PLDA speaker recognition platform. This is in no means a comprehensive introduction for this type of system, rather its function is to provide context for proposed systems in Chapter 4. Speaker recognition in i-Vector/PLDA system is implemented in a way to fall in the speaker verification framework. Much like a GMM-UBM system, each decision provides a likelihood ratio verifying whether a train speaker matches audio from a test speaker. An i-Vector/PLDA system typically trains a speaker model (i.e., i-Vector) using multiple training files. Unlike GMM-UBM, test speakers are also represented by speaker models. Therefore, train and test speakers are treated similarly in an i-Vector/PLDA system. Figure A.1 summarizes a typical i-Vector/PLDA system (Dehak et al., 2011).

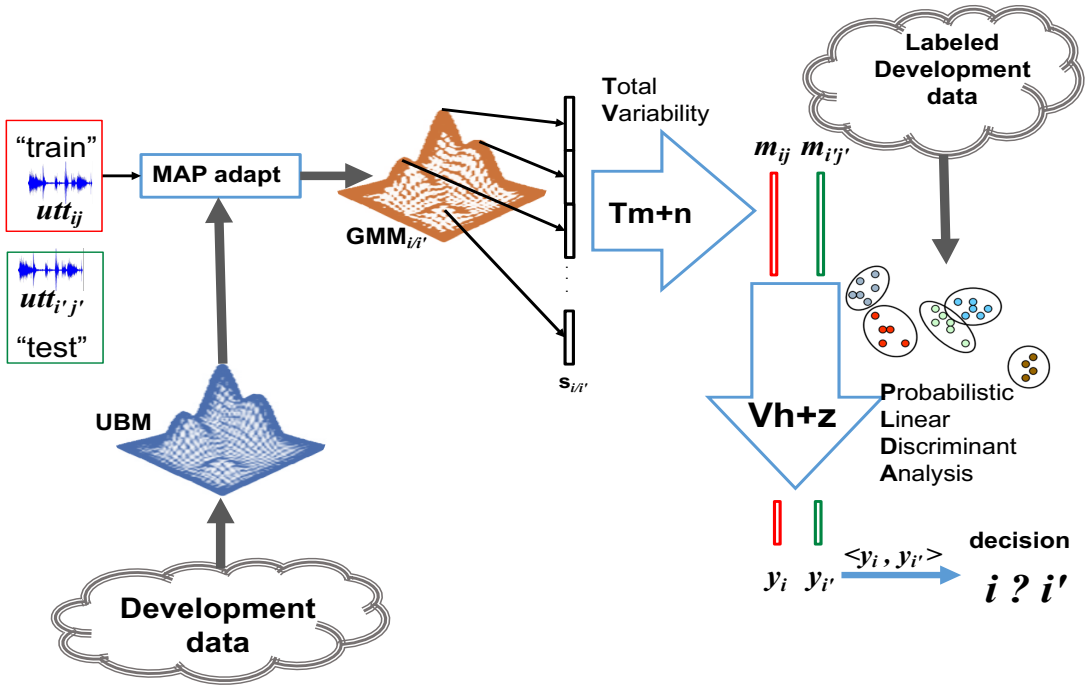


Figure A.1. Overview of i-Vector/PLDA system.

The first step is extract to i-Vectors from a collection of audio files (or a single audio file), provided for each speaker. i-Vectors are obtained by compressing Gaussian mean super-vectors. Super-vectors are a concatenation of GMM means. Since GMMs are typically composed of 512 (or more) Gaussian mixtures, the concatenated GMM means result in incredibly large super-vectors (36-dimensional MFCC means $\times$ 512 Gaussian mixtures). Therefore, i-Vectors can be viewed as a dimension reduction technique to reduce the dimension of super-vectors. This is done through a factor analysis process that represents super-vectors as factor loadings of a Total Variability (TV) matrix. The factor loading weights are called i-Vectors (denoted $\mathbf{m}$ in Fig. A.1). TV matrix can be trained using unlabeled development data (typically the same data used to train the UBM).

An i-Vector is provided for the train and test components in each trial. The next step is to compare these two i-Vectors to determine whether they represent the same speaker. Many methods have been proposed to compare i-Vectors, the most popular of which is probabilistic linear discriminant analysis (PLDA). PLDA has been thoroughly explained in Chapter 4, but it can also be viewed as a factor analysis method to remove certain (unwanted) variabilities from the i-Vectors. The likelihood ratio provided by PLDA is typically used to determine the certainty to which the system believes the two i-Vectors from a trial belong to the same speaker.

## REFERENCES


Angkititrakul, P., M. Petracca, A. Sathyanarayana, and J. H. Hansen (2007). Utdrive: driver behavior and speech interactive systems for in-vehicle environments. In *2007 IEEE Intelligent Vehicles Symposium*, pp. 566–569. IEEE.

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

Boakye, K., , O. Vinyals, and G. Friedland (2008). Two's a crowd: Improving speaker diarication by automatically identifying and excluding overlapped speech. In *Proc. INTERSPEECH*, Brisbane, Australia, pp. 32–35.

Boakye, K. (Fall 2008). *Audio Segmentation for Meeting Speech Processing*. Ph. D. thesis.

Boakye, K., B. Trueba-Hornero, O. Vinyals, and G. Friedland (2008). Overlapped speech detection for improved diarization in multiparty meetings. In *Proc. IEEE ICASSP-2008: Inter. Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, NV, pp. 4353–4356.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing 27*(2), 113–120.

Bredin, H. and J. Poignant (2013). Integer linear programming for speaker diarization and cross-modal identification in tv broadcast. In *the 14rd Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Brümmer, N. and E. De Villiers (2010). The speaker partitioning problem. In *Odyssey*, pp. 34.

Brümmer, N. and E. de Villiers (2013). The bosaris toolkit: Theory, algorithms and code for surviving the new dcf. *arXiv preprint arXiv:1304.2865*.

Burget, L., O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brummer (2011, May). Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 4832–4835.

Carletta, J., S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner (2006). The ami meeting corpus: A pre-announcement. In *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, MLMI'05, Berlin, Heidelberg, pp. 28–39. Springer-Verlag.

Carlson, A. B. and P. B. Crilly (2010). Communication systems, 5e.

Cetin, O. and E. Shriberg (2006, May). Speaker overlaps and asr errors in meetings: Effects before, during, and after the overlap. In *Proc. IEEE ICASSP-2006: Int. Conf. Acoustics, Speech, and Signal Processing*, Toulouse, France, pp. 357–360.

Cettolo, M. and M. Vescovi (2003). Efficient audio segmentation algorithms based on the bic. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, Volume 6, pp. VI–537. IEEE.

Cettolo, M., M. Vescovi, and R. Rizzi (2005). Evaluation of bic-based algorithms for audio segmentation. *Computer Speech & Language 19*(2), 147–170.

Chazan, D., Y. Stettiner, and D. Malah (1993, April). Optimal multipitch estimation using the em algorithm for co-channel speech separation. In *Proc. IEEE ICASSP-93: Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 728–731.

Chen, S. and P. Gopalakrishnan (1998). Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Volume 8, pp. 127–132. Virginia, USA.

Cohen, L. and C. Lee (1990, Apr). Instantaneous bandwidth for signals and spectrogram. In *Proc. IEEE ICASSP-90: Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 2451–2454 vol.5.

Cooke, M., J. Barker, S. Cunningham, and X. Shao (2006, November). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America 120*(5), 2421–2424.

Cooke, M., J. R. Hershey, and S. J. Rennie (2010). Monaural speech separation and recognition challenge. *Computer Speech and Language 24*(1), 1 – 15.

Cooke, M. and T. Lee. Speech separation challenge.

Cumani, S., O. Plchot, and P. Laface (2013, May). Probabilistic linear discriminant analysis of i-vector posterior distributions. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 7644–7648.

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

Dongxin, X., U. Yapanel, S. Gray, J. Gilkerson, J. Richards, and J. Hansen (2008). Signal processing for young child speech language development. In *Workshop on Child Computer Interaction (WOCCI)*.

Dupuy, G., S. Meignier, P. Deléglise, and Y. Esteve (2014). Recent improvements on ilp-based clustering for broadcast news speaker diarization. In *Proceedings of Odyssey*. Citeseer.

Dupuy, G., M. Rouvier, S. Meignier, and Y. Esteve (2012). I-vectors and ilp clustering adapted to cross-show speaker diarization. In *INTERSPEECH*, pp. 2174–2177.

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

Giuliani, M., T. L. Nwe, and H. Li (2006). Meeting segmentation using two-layer cascaded subband filters. In *ISCSLP'06*, pp. 672–682.

Greenberg, C. S., D. Bans, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, and D. A. Reynolds (2012, Jun.). The nist 2014 speaker recognition i-vector machine learning. In *Proc. ISCA Odyssey*, Singapore, Singapore.

Hansen, J. H. and T. Hasan (2015). Speaker recognition by machines and humans: a tutorial review. *IEEE Signal Processing Magazine 32*(6), 74–99.

Hasan, T., S. O. Sadjadi, G. Liu, N. Shokouhi, H. Bořil, and J. H. Hansen (2013). Crss systems for 2012 nist speaker recognition evaluation. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6783–6787. IEEE.

Hasan, T., R. Saeidi, J. H. Hansen, and D. A. van Leeuwen (2013). Duration mismatch compensation for i-vector based speaker recognition systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7663–7667. IEEE.

Hu, X., S. Peng, and W.-L. Hwang (2012). Multicomponent AM-FM signal separation and demodulation with null space pursuit. *Signal, Image and Video Processing*, 1–10.

Ioffe, S. (2006). Probabilistic linear discriminant analysis. In *European Conference on Computer Vision*, pp. 531–542. Springer.

Kaiser, J. (1990, Apr). On a simple algorithm to calculate the 'energy' of a signal. In *Proc. IEEE ICASSP-90: Inter. Conf. Acoustics, Speech, and Signal Processing*, pp. 381–384 vol.1.

Kaiser, J. F. (1993, April). Some useful properties of teager's energy operators. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Volume 3, pp. 149–152 vol.3.

Kelly, F., A. Drygajlo, and N. Harte (2013). Speaker verification in score-ageing-quality classification space. *Computer Speech and Language 27*(5), 1068 – 1084.

Kelly, F. and J. H. Hansen (2016). Score-aging calibration for speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Kenny, P. (2010). Bayesian speaker verification with heavy-tailed priors. In *Odyssey*, pp. 14.

Krishnamachari, K., R. E. Yantorno, D. S. Benincasa, and S. J. Wenndt (2000, November). Spectral autocorrelation ratio as a usability measure of speech segments under co-channel conditions. In *IEEE Intl. Symp. on Intelligent Signal Processing and Communication Systems, ISPACS*, pp. 710–713.

Krishnamachari, K., R. E. Yantorno, J. M. Lovekin, D. S. Benincasa, and S. J. Wenndt (2001). Use of local kurtosis measure for spotting usable speech segments in co-channel speech. In *Proc. IEEE ICASSP-2001: Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, pp. 649–652.

Kryszczuk, K. and A. Drygajlo (2009). Improving biometric verification with class-independent quality information. *IET Signal Process., IEEE 3*, 310–321.

LeBlanc, J. and P. De Leon (1998, May). Speech separation by kurtosis maximization. In *Proc. IEEE ICASSP-98: Int. Conf. Acoustics, Speech, and Signal Processing*, Volume 2, pp. 1029–1032 vol.2.

Lei, Y., L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer (2012, March). Towards noise-robust speaker recognition using probabilistic linear discriminant analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4253–4256.

Lin, W., C. Hamilton, and P. Chitrapu (1995, May). A generalization to the teager-kaiser energy function and application to resolving two closely-spaced tones. In *Proc. IEEE ICASSP-95: Inter. Conf. Acoustics, Speech, and Signal Processing*, Volume 3, pp. 1637–1640 vol.3.

Litvin, Y., I. Cohen, and D. Chazan (2010). Monaural speech/music source separation using discrete energy separation algorithm. *Signal Processing 90*(12), 3147 – 3163.

Lovekin, J., K. R. Krishnamachari, R. E. Yantorno, D. S. Benincasa, and S. J. Wenndt (2001, April). Adjacent pitch period comparison (appc) as a usability measure of speech segments under co-channel conditions.

Makhorin, A. (2008). Glpk (gnu linear programming kit).

Mandasari, M., R. Saeidi, M. McLaren, and D. van Leeuwen (2013, Nov). Quality measure functions for calibration of speaker recognition systems in various duration conditions. *Audio, Speech, and Language Processing, IEEE Transactions on 21*(11), 2425–2438.

Maragos, P., J. Kaiser, and T. Quatieri (1993, Oct). Energy separation in signal modulations with application to speech analysis. *Signal Processing, IEEE Transactions on 41*(10), 3024–3051.

Maragos, P., A. Potamianos, R. Potamianos, B. Santhanam, and G. Xx (1995). Instantaneous energy operators: Applications to speech processing and communications. In *IEEE Workshop on Nonlinear Signal and Image Proc.*, Thessaloniki, Greece.

Matejka, P., O. Glembek, F. Castaldo, M. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky (2011, May). Full-covariance ubm and heavy-tailed plda in i-vector speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 4828–4831.

Meignier, S. and T. Merlin (2010). Lium spkdiarization: an open source toolkit for diarization. In *CMU SPUD Workshop*, Volume 2010.

Miro, X. A. (2007). *Robust speaker diarization for meetings*. Universitat Politècnica de Catalunya.

Misra, A. and J. H. Hansen (2014). Spoken language mismatch in speaker verification: An investigation with nist-sre and crss bi-ling corpora. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pp. 372–377. IEEE.

Morgan, D. P., E. B. George, L. T. Lee, and S. M. Kay (1997, Sep). Cochannel speaker separation by harmonic enhancement and suppression. *IEEE Transactions on Speech and Audio Processing 5*(5), 407–424.

Mowlaee, P., R. Saeidi, Z.-H. Tan, M. G. Christensen, P. Fränti, and S. H. Jensen (2010). Joint single-channel speech separation and speaker identification. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4430–4433. IEEE.

Mueller, M. and S. Kramer (2010). Integer linear programming models for constrained clustering. In *International Conference on Discovery Science*, pp. 159–173. Springer.

Ning, H., M. Liu, H. Tang, and T. S. Huang (2006). A spectral clustering approach to speaker diarization. In *INTERSPEECH*.

NIST (2004). The NIST year 2004 speaker recognition evaluation plan.

NIST (2005). The NIST year 2005 speaker recognition evaluation plan.

NIST (2006). The NIST year 2006 speaker recognition evaluation plan.

NIST (2008). The NIST year 2008 speaker recognition evaluation plan.

Panahi, I. M. and K. Venkat (2009). Blind identification of multi-channel systems with single input and unknown orders. *Signal Processing 89*(7), 1288–1310.

Potamianos, A. and P. Maragos (1995, May). Speech formant frequency and bandwidth tracking using multiband energy demodulation. In *Proc. IEEE ICASSP-95: Int. Conf. Acoustics, Speech, and Signal Processing*, Volume 1, pp. 784–787 vol.1.

Potamianos, A. and P. Maragos (1996, June). Speech formant frequency and bandwidth tracking using multiband energy demodulation. *J. Acoust. Soc. Amer. 99*(6), 3795–3806.

Povey, D., A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, Number EPFL-CONF-192584. IEEE Signal Processing Society.

Prazak, J. and J. Silovsky (2011). Speaker diarization using plda-based speaker clustering. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, Volume 1, pp. 347–350. IEEE.

Prince, S. and J. Elder (2007, Oct). Probabilistic linear discriminant analysis for inferences about identity. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8.

Quatieri, T. and R. Danisewicz (1990, January). An approach to co-channel talker interference suppression using a sinusoidal model for speech. *IEEE Trans. Acoustics Speech and Signal Process. 3X*(I), 56–69.

Reynolds, D., T. Quatieri, and R. Dunn (2000, September). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing 10*, 19–41.

Rossignol, S. and O. Pietquin (2010). Single-speaker/multi-speaker co-channel speech classification. In *INTERSPEECH*, pp. 2322–2325.

Rouvier, M. and S. Meignier (2012). A global optimization framework for speaker diarization. In *Odyssey*, pp. 146–150.

Sadjadi, S., M. Slaney, and L. Heck (2013, November). MSR identity toolbox v1.0: A matlab toolbox for speaker-recognition research. *Speech and Language Processing Technical Committee Newsletter*.

Sadjadi, S. O. and J. H. Hansen (2013). Unsupervised speech activity detection using voicing measures and perceptual spectral flux. *IEEE Signal Processing Letters 20*(3), 197–200.

Sadjadi, S. O. and L. P. Heck (2014). Speaker verification based processing for robust ASR in co-channel speech scenarios. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pp. 1774–1778.

Saeidi, R., P. Mowlaee, T. Kinnunen, Z.-H. Tan, M. G. Christensen, S. H. Jensen, and P. Franti (2010). Signal-to-signal ratio independent speaker identification for co-channel speech signals. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 4565–4568. IEEE.

Santhanam, B. and P. Maragos (2000, Mar). Multicomponent AM-FM demodulation via periodicity-based algebraic separation and energy-based demodulation. *Communications, IEEE Transactions on 48*(3), 473–490.

Sathyanarayana, A., S. O. Sadjadi, and J. H. L. Hansen (2012a, September). Levaraging sensor information from portable devices towards automatic driving maneuver recognition. In *IEEE 15th Intl. Conference on Intelligent Transportation Systems*, Anchorage, AK.

Sathyanarayana, A., S. O. Sadjadi, and J. H. L. Hansen (2012b, January). Leveraging speech-active regions towards active safety in vehicles. In *IEEE Intl. Conf. Emerging Signal Processing Applications, ESPA 2012*, Las Vegas, pp. 48–51.

Sathyanarayana, A., S. O. Sadjadi, and J. H. L. Hansen (2013, April). Automatic driving maneuver recognition and analysis using cost effective portable devices. In *SAE World Congress abd Exhibition*, Detroit.

Sathyanarayana, A., N. Shokouhi, S. O. Sadjadi, and J. H. Hansen (2013). Belt up: Investigating the impact of in-vehicular conversation on driving performance. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pp. 1071–1076. IEEE.

Schegloff, E. A. (2000). Overlapping talk and the organization of turn-taking for conversation. *Language in society 29*(01), 1–63.

Schegloff, E. A. (2002). *Accounts of Conduct in Interaction: Interruption, Overlap and turn-taking, in J. H. Turner (ed.), Handbook of Sociolinguistics*, pp. 287–321. New York: Plenum.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics 6*(2), 461–464.

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

Sell, G. and D. Garcia-Romero (2015). Diarization resegmentation in the factor analysis subspace. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4794–4798. IEEE.

Shao, Y. and D. L. Wang (2003). Co-channel speaker identification using usable speech extraction based on multi-pitch tracking. In *Proc. IEEE ICASSP-2003: Inter. Conf. Acoustics, Speech, and Signal Processing*, Hong Kong, pp. 205–208.

Shokouhi, N. and J. H. Hansen (2015). Probabilistic linear discriminant analysis for robust speaker identification in co-channel speech. In *Sixteenth Annual Conference of the International Speech Communication Association, INTERSPEECH*, Dresden, Germany.

Shokouhi, N. and J. H. L. Hansen (2016). Teager-kaiser energy operators for overlapped speech detection. *submitted*.

Shokouhi, N., A. Sathyanarayana, S. Sadjadi, and J. H. L. Hansen (2013, May). Overlapped-speech detection with applications to driver assessment for in-vehicle active safety systems. In *Proc. IEEE ICASSP-2013: Inter. Conf. Acoustics, Speech, and Signal Processing*, Vancouver, BC.

Shokouhi, N., A. Ziaei, A. Sangwan, and J. H. L. Hansen (2015, April). Robust overlapped speech detection and its application in word-count estimation for prof-life-log data. In *Proc. IEEE ICASSP-2015: Inter. Conf. Acoustics, Speech, and Signal Processing*, Brisbane, Australia.

Shriberg, E., A. Stolcke, and D. Baron (2001). Observations on overlap: Findings and implications for automatic processing of multi-party conversation. In *in Proceedings of Eurospeech*, pp. 1359–1362.

Shum, S., N. Dehak, E. Chuangsuwanich, D. A. Reynolds, and J. R. Glass (2011). Exploiting intra-conversation variability for speaker diarization. In *INTERSPEECH*, Volume 11, pp. 945–948.

Shum, S., N. Dehak, and J. Glass (2012). On the use of spectral and iterative methods for speaker diarization. *System 1*(w2), 2.

Silovsky, J., J. Prazak, P. Cerva, J. Zdansky, and J. Nouza (2011). Plda-based clustering for speaker diarization of broadcast streams. In *Twelfth Annual Conference of the International Speech Communication Association*.

Sizov, A., K. A. Lee, and T. Kinnunen (2014). Unifying probabilistic linear discriminant analysis variants in biometric authentication. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 464–475. Springer.

Smolenski, B. and R. Ramachandran (2011). Usable speech processing: A filterless approach in the presence of interference. *Circuits and Systems Magazine, IEEE 11*(2), 8 –22.

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

Wang, D. and S. Narayanan (2007). Robust speech rate estimation for spontaneous speech. *IEEE Transactions on Audio, Speech, and Language Processing 15.8*, 2190–2201.

Wrigley, S. N., G. J. Brown, W. Vincent, and S. Renals (2005, January). Speech and crosstalk detection in multichannel audio. *IEEE Trans. Audio Speech Lang. Process. 13*(1), 84–91.

Wu, M., D. L. Wang, and G. J. Brown (2003, May). A multi-pitch tracking algorithm for noisy speech. *IEEE Trans. on Speech and Audio Process. 11*, 229–241.

Xiao, B., P. K. Ghosh, P. Georgiou, and S. S. Narayanan (2011). Overlapped speech detection using long-term spectro-temporal similarity in stereo recording. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5216–5219. IEEE.

Yantorno, R. E. (September 1999). Cochannel speech study. Technical report, Electrical and Computer Engineering Department Temple University.

Yantorno, R. E., D. S. Benincasa, and S. J. Wenndt (2000, November). Effects of co-channel speech on speaker identification. In *SPIE Intl. Symp. on Tech. for Law Enforcement*.

Zelenák, M., H. Schulz, and F. J. Hernando Pericás (2010). Albayzin 2010 evaluation campaign: speaker diarization. In *VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop*, pp. 301–304.

Zelenak, M., C. Segura, J. Luque, and J. Hernando (2012, Feb). Simultaneous speech detection with spatial features for speaker diarization. *Audio, Speech, and Language Processing, IEEE Transactions on 20*(2), 436–446.

Zhao, X., Y. Wang, and D. Wang (2015). Cochannel speaker identification in anechoic and reverberant conditions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 23*(11), 1727–1736.

Zhou, B. and J. H. Hansen (2005). Efficient audio stream segmentation via the combined t 2 statistic and bayesian information criterion. *IEEE Transactions on Speech and Audio Processing 13*(4), 467–474.

Ziaei, A., A. Sangwan, and J. H. L. Hansen (2013). Prof-life-log: Personal interaction analysis for naturalistic audio streams. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7770–7774. IEEE.

Ziaei, A., A. Sangwan, and J. H. L. Hansen (2014, September). A speech system for estimating daily word counts. In *Proc. INTERSPEECH*, Singapore.

## BIOGRAPHICAL SKETCH

Navid Shokouhi was born in Ahvaz, Khuzestan, Iran on June 9, 1989, to Hossein Shokouhi and Manzar Mohammadi. His parents decided to move to Australia soon after he was born. At the age of 9 he and his parents moved back to Ahvaz, where his father was hired as an Assistant Professor of Linguistics in Shahid Chamran University. Navid attended Amirkabir University of Technology (AUT) in Tehran as a Bachelors student in Electrical Engineering in 2007. He graduated with a Degree in Electrical Engineering in 2011. As a senior undergraduate student he worked with Dr. Hamid Sheikhzadeh for his final project. There he was introduced to basic principles of speech signal processing. His undergraduate senior project at AUT was on voice conversation and speech synthesis. In the Fall of 2011 he moved to the United States to pursue a PhD degree under the supervision of Professor John H. L. Hansen at The University of Texas at Dallas (UTD). In the Center for Robust Speech Systems (CRSS) at UTD he primarily focused on speaker recognition, specifically in multi-speaker environments. He participated in a number of NIST Speaker Recognition Evaluations during his time as a PhD student at CRSS. In 2013 he was awarded the IBM best student paper award by the IEEE signal processing society alongside three other student co-authors for a paper in ICASSP 2013 on the CRSS system for speaker recognition in the 2012 NIST SRE challenge. In the Summer of 2015 he was recruited as an intern at ToyTalk Inc., where he developed speech processing tools for mobile platforms. He is expected to graduate with a PhD degree in Electrical Engineering – Signal Processing from The University of Texas at Dallas in 2016.

# CURRICULUM VITAE

**Navid Shokouhi**

424 Dorset Rd.,
Croydon,
Melbourne, VIC 3136
Australia

*Phone:* +1 (469) 394-6868
*E-mail:* navid.shokouhi@utdallas.edu
*web:* www.linkedin.com/in/navidshokouhi

## Research Interests

Robust speech/speaker recognition,
Acoustic Modeling, Machine learning

## Education

**The University of Texas at Dallas**,

Ph.D. Candidate, EE, Fall 2011 - present
Expected graduation date: December 2016
- Dissertation Topic: "Speaker Recognition and Diarization in Co-channel Speech"
  *Addresses tangible solutions for speaker recognition*
  *in multi-speaker speech signals (e.g. meetings, conversations, etc.)*
- Advisor: Dr. John H. L. Hansen

**Amirkabir University of Technology**, Tehran, Iran

B.Eng., Electrical Engineering

## Experience

**Intern** June 2015 - Aug 2015
ToyTalk Inc.
*Speech engineering intern*
*Built online/offline voice activity detection .*

**Research Assistant** September 2011 - Present
The University of Texas at Dallas

## Honors and Awards

**ICASSP 2013 best student paper award** February 2013
IEEE Signal Processing Society on behalf of IBM

**Undergraduate Honors student** 2011
Amirkabir University of Technology, Tehran, Iran
Exemption from graduate school entrance exam

## Projects

**2012 NIST Speaker Recognition Evaluation** June 2012 - October 2012
*Managed the fusion/calibration system for the UTDallas SRE submission.*

**2014 NIST i-vector challenge** December 2013

**2016 NIST Speaker Recognition Evaluation** October 2016

**CRSS-SpkrDiar** 2016
*Developed C++ library for end-to-end speaker diarization.*

## Skills

- Python, C++, MATLAB: use on a daily basis.
- Perl and Bash scripting: basic familiarity for text processing.

## Test Scores

**Iranian Regional Olympiads for University Students - Electrical Eng.** March 2010
Tehran, Iran
*Ranked 8th*
**Iranian National Olympiads for University Students - Electrical Eng.** June 2010
Tehran, Iran
*Ranked 20th*

## Volunteer Work

**IEEE SLT Committee Newsletter** January 2013 - 2016
News staff reporter

**SLT 2014 organizing committee - lead student volunteer** December 2014

## Publications

**Navid Shokouhi**, John H. L. Hansen, "Teager-Kaiser Energy Operators for Overlapped Speech Detection, " submitted to IEEE Trans. on ASLP (accepted – to be published in 2017).

**Navid Shokouhi**, John H. L. Hansen, "Speaker Recognition in Co-channel Speech Using Modified Probabilistic Linear Discriminant Analysis, " JP – (under revision).

John H. L. Hansen, Mahesh Kumar Nandwana, **Navid Shokouhi**, "Analysis of human scream and its impact on text-independent speaker veri
cation, " submitted to JASA (under revision).

Yang Zheng, **Navid Shokouhi**, Nicolai Thomsen, Amardeep Sathyanarayana, John H. L. Hansen, "Towards Developing a Distraction-Reduced Hands-Off Interactive Driving Experience using Portable Smart Devices, " SAE Technical Paper, January 2016.

**Navid Shokouhi**, Yang Zheng, Amardeep Sathyanarayana, John H. L. Hansen, "In-Vehicle Conversation Analysis towards Improved Driver Assistance Systems," 7th Biennial Workshop on DSP for In-Vehicle Systems and Safety, Berkeley, CA, October 2015.

**Navid Shokouhi**, John H. L. Hansen, "Probabilistic Linear Discriminant Analysis for Robust Speaker Identification in Co-channel Speech," 16th Annual Conference of the International Speech Communication Association (INTERSPEECH), Dresden, Germany, Sept. 6-10, 2015.

**Navid Shokouhi**, Ali Ziaei, Abhijeet Sangwan, John H. L. Hansen, "Robust Overlapped Speech Detection and its Applications in Word-Count Estimation for Prof-Life-Log Data," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, April 2015.

**Navid Shokouhi**, Seyed Omid Sadjadi, John H. L. Hansen, "Co-channel Speech Detection via Spectral Analysis of Frequency Modulated Sub-bands," 15th Annual Conference of the International Speech Communication Association (INTERSPEECH), Singapore, September 2014.

Gang Liu, Chengzhu Yu, Abhinav Misra, **Navid Shokouhi**, John H. L. Hansen , "Investigating State-of-the-Art Speaker Verification in the Case of Unlabeled Development Data," Odyssey: The Speaker and Language Recognition Workshop, Finland, 2014.

Jamal Amini, Abdoreza Sabzi Shahrebabaki, **Navid Shokouhi**, Hamid Sheikhzadeh, Kamran Raahemifar, Mahdi Eslami, "Speech analysis/synthesis by Gaussian mixture approximation of the speech spectrum for voice conversion" IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Greece, December 2013.

Amardeep Sathyanarayana, **Navid Shokouhi**, Seyed Omid Sadjadi, John H. L. Hansen, "Belt Up: Investigating the impact of in-vehicular conversation on driving performance," Intelligent Vehicles Symposium (IV), IEEE, Australia, June 2013.

Taufiq Hasan, Seyed Omid Sadjadi, Gang Liu, **Navid Shokouhi**, Hynek Boril, John H. L. Hansen, "CRSS systems for 2012 NIST speaker recognition evaluation," IEEE International

Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, May 2013.

**Navid Shokouhi**, Amardeep Sathyanarayana, Seyed Omid Sadjadi, John H. L. Hansen, "Overlapped-speech detection with applications to driver assessment for in-vehicle active safety systems," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, May 2013.

**Navid Shokouhi**, Amardeep Sathyanarayana, Seyed Omid Sadjadi, John H. L. Hansen, "Analysis of In-Vehicle Speech Activity towards Driver Safety Assessment," 6th Biennial Workshop on DSP for In-Vehicle Systems and Safety, Seoul, South Korea, 2013.

**Navid Shokouhi** among others, "I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification," 14th Annual Conference of the International Speech Communication Association (INTERSPEECH), Lyon, France, August 2013.

Online Articles John H. L. Hansen, **Navid Shokouhi**, "Speaker Identification: Screaming, Stress and Non-Neutral Speech, is there speaker content?" IEEE Speech and Language Technical Committee (SLTC) Newsletter, November 2013.

Taufiq Hasan, Gang Liu, Seyed Omid Sadjadi, **Navid Shokouhi**, H Boil, A Ziaei, Abhinav Misra, KW Godin, John H. L. Hansen, "UTD-CRSS systems for 2012 NIST speaker recognition evaluation," Proc. NIST SRE Workshop, December 2012.

Chunlei Zhang, Fahimeh Bahmaninezhad, Shivesh Ranjan, Chengzhu Yu, **Navid Shokouhi**, John H. L. Hansen. "UTD-CRSS Systems for 2016 NIST Speaker Recognition Evaluation, " arXiv preprint arXiv:1610.07651, October 2016.