DEEP NEURAL NETWORKS AND MODEL-BASED APPROACHES FOR ROBUST SPEAKER DIARIZATION IN NATURALISTIC AUDIO STREAMS

by

Harishchandra Dubey

APPROVED BY SUPERVISORY COMMITTEE:

Dr. John H. L. Hansen, Chair

Dr. Abhijeet Sangwan

Dr. Carlos Busso

Dr. P. K. Rajasekaran

Dr. Chin-Tuan Tan

Copyright © 2019

Harishchandra Dubey

All rights reserved

To my parents and my wife, Ankita.

DEEP NEURAL NETWORKS AND MODEL-BASED APPROACHES FOR ROBUST SPEAKER DIARIZATION IN NATURALISTIC AUDIO STREAMS

by

HARISHCHANDRA DUBEY, B.TECH, M.Sc.

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

August 2019

ACKNOWLEDGMENTS

I would like to extend my heartfelt thanks to my advisor, Dr. John H. L. Hansen, for his dedication, support and advice throughout my PhD program. This dissertation was possible only with his systematic advice and encouragement during the course of my PhD studies. My immense learning at CRSS is largely carved out by stable financial support that allowed me to go beyond one topic and dive deeper in robust speech systems. I consider myself lucky to have been a part of CRSS which will make significant impact on the rest of my life. I would like to specially thank Dr. Abhijeet Sangwan for being a great support, especially in first year of my PhD studies that helped me in learning the tools and techniques related to my PhD project. Later years were more productive mostly out of the discipline I learned in my first year. I would like to express gratitude and appreciation to my committee members: Dr. P. K. Rajasekaran, Dr. Carlos Busso, and Dr. Chin-Tuan Tan for their suggestions on research methodologies and presentation. Special thanks to Tammy Emery and Rosarita Lubag for support with administrative processing related to degree and course completion formalities. I would like to thank the students and staff at CRSS throughout these years for helpful comments and discussions related to research, coding, and paper writing. Sharing an office with Chunlei Zhang, Shivesh Ranjan and Abhinav Misra in one PhD program provided me a great balance of different perspectives and also motivation for working long hours every day. I thank Lakshmish Kaushik and Chengzhu Yu for advice on efficient coding skills that helped me immensely throughout my PhD program. I thank Gang Liu for his advice and suggestions. CRSS turned out to be an awesome learning experience that I will cherish forever. I have learned more from my lab mates and advisors than from formal classroom teaching. Dr. Hansen's kindness for letting all students to attend conferences not only helped in networking but also inspired to maintain high standard on research efforts. Last but not least, I thank my grandfather, a kind soul who as a mathematics professor inspired me to pursue my dreams towards this academic journey. I am grateful to my parents, Arun Kumar

Dubey and Prem Kumari Dubey, for their unconditional love, trust, and sacrifices that held me high in tough times. Finally, I would like to thank my wife, Ankita Dubey, for her love, understanding and support.

May 2019

DEEP NEURAL NETWORKS AND MODEL-BASED APPROACHES FOR ROBUST SPEAKER DIARIZATION IN NATURALISTIC AUDIO STREAMS

Harishchandra Dubey, PhD
The University of Texas at Dallas, 2019

Supervising Professor: Dr. John H. L. Hansen, Chair

Speaker diarization is an unsupervised task that determines "who spoke and when" within input audio stream. It consists of four sub-systems: (i) speech activity detection (SAD); (ii) speaker segmentation and modeling; (iii) speaker clustering and (iv) re-segmentation. Previous diarization systems have addressed telephone and/or meeting recordings in cleaner, but fail in naturalistic audio streams. Naturalistic audio such as CRSS-PLTL corpus consists of short-speaker turns and distortions including noise, reverberation, overlapped speech, and other miscellaneous human non-speech vocalizations. These factors pose challenge for speaker diarization in naturalistic audio. This dissertation formulates several systems to enhance speaker diarization, resulting in four contributions. The first contribution advances SAD based on frequency-dependent kernel (FDK-SAD) features and three alternate decision backends, namely: (i) Variable model-size GMM (VMGMM), (ii) Hartigan dip test based robust feature clustering (DipSAD), and (iii) Cumulative density based linear curve (D-SAD). Evaluations employ open-source corpora such as NIST OpenSAD-2015, NIST OpenSAT-2017, redDots and CRSS-PLTL corpus. CRSS-PLTL contains multi-stream audio recordings from UTDallas student-led STEM teaching model . Second, novel architectures are developed based on SincNet convolutional neural network for speaker identification and diarization. Proposed models generalize well with smaller training data, hence an attractive choice for

transfer learning (TL). The standard SincNet architecture is expanded by introducing both additive margin (AM)-Softmax and Center loss, which leads to four novel architectures namely (i) standard SincNet, (ii) AM-SincNet, (iii) AM-CL-SincNet, and (iv) CL-SincNet for speaker diarization. We leverage transfer learning (TL) for training SincNet models on two training datasets: (1) TIMIT, (2) Librispeech corpus. Diarization evaluations are conducted on UT Dallas CRSS-PLTL and AMI meeting corpora. Thirdly, three novel algorithms are proposed for speaker clustering: (i) Mixture of von Mises-Fisher distributions (movMF); (ii) Normalized Fuzzy C-means clustering (NFCM); and (iii) Toeplitz Inverse Covariance-based speaker clustering (TIC). While TIC is computationally complex than movMF and NFCM, it out-performs both movMF and NCFM. Finally, speech systems are proposed for knowledge extraction and interaction analysis using unsupervised or pre-trained models, using Peer-Led Team Learning (PLTL) sessions. We leverage CRSS Speech Profiler for detecting four low-level attributes namely: (i) Emotion recognition; (ii) Lombard effect; (iii) Whisper detection; and (iv) Physical task stress. These low-level attributes are used for unsupervised PLTL interaction analysis aimed at assessing student engagement. The resulting evaluations of both publically available corpora, as well as UT Dallas PLTL data, confirm the impact of the proposed algorithmic advancements for diarization in naturalistic audio streams. Taken collectively, the resulting dissertation contributions advance a number of processing sub-tasks to achieve effective robust speaker diarization in naturalistic streams.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AIC Akaike information criterion

ARMA Auto-Regressive Moving Average

AM Additive Margin

ASR Automatic Speech Recognition

BIC Bayesian information criterion

CRSS Center for Robust Speech Systems

CL Center Loss

CNN Convolutional Neural Network

DARPA Defense Advanced Research Projects Agency

DCF Detection Cost Function

DNN Deep Neural Network

DER Diarization Error Rate

DipSAD Dip-based robust feature clustering

eng American English

EM Expectation Maximization

FDK Frequency-Dependent Kernel

FDGK Frequency-Dependent Gaussian Kernel

GMM Gaussian Mixture Model

GLR Generalized Likelihood Ratio

IIR Infinite Impulse Response

HMM Hidden Markov Model

alv Levantine Arabic

LSTM Long Short-term Memory Neural Network

ML Maximum Likelihood

ML Maximum Likelihood

MAP maximum-a-posteriori

MLE Maximum Likelihood Estimation

MVN Mean and Variance Normalization

MFCC Mel-Frequency Cepstral Coefficient

MLP Multilayer Perceptron

MO-LRT Multiple Observation Likelihood Ratio Test

MI Mutual Information

NIST National Institute of Standards and Technology

OpenSAT NIST Speech Analytic Technologies evaluation series

PLTL Peer-Led Team Learning

PCA Principal Component Analysis

PDF Probability Density Function

LSF Line Spectral Frequencies

LL Log-Likelihood

LDNN Long-duration Naturalistic Noise

PSC Public Safety Communications

RATS Robust Automatic Transcription of Speech

SAD Speech Activity Detection

SNR Signal to Noise Ratio

SincNet Sinc Convolutional Neural Network

SO-LRT Single Observation Likelihood Ratio Test

SSGMM Semi-supervised Gaussian Mixture Model

SRMR speech-to-reverberation modulation energy ratio

SAcC sub-band autocorrelation

SVM support vector machine

UBM Universal Background Model

urd Urdu

VMGMM Variable model-size Gaussian mixture model

VTL Vanilla Transfer Learning

vMF von Mises-Fisher

WPE Weighted Prediction Error

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Dissertation Motivation

The task of speaker diarization addresses the basic question "who spoke and when" within an audio stream (Anguera et al., 2012; Tranter and Reynolds, 2006). Speaker recognition, which tries to identify speakers from a closed set. Both tasks are equivalent except for two differences: (i) diarization involves speaker clustering (unsupervised) while recognition is a classification task (supervised); (ii) speaker recognition assumes the availability of enrollment data for each speaker, unlike diarization which has no enrollment data (Hansen and Hasan, 2015). Speaker recognition and diarization are important tasks for practical audio stream applications such as voice authentication, multi-speaker speech recognition, interaction analysis, meeting annotations, or audio retrieval. Commercial voice assistants such as Microsoft Cortana, Apple Siri, Amazon Alexa, and Google Home employ speaker diarization and recognition for delivering personalized voice and speech services. Speaker diarization can in general be used as a front-end for analysis of meeting conversations such as the AMI corpus (Carletta et al., 2005). Alternatively,another multi-subject audio corpus is derived from Peer-Lead Team Learning (PLTL) which is a student-led STEM education model popular in US universities. It is a structured program where a team leader facilitates collaborative problem solving among a small-group of students. Another important application of speaker diarization is interaction analysis, such as exploring individual student engagement within PLTL sessions. PLTL sessions have been shown to improve student learning that lead to improvement in their grades (Snyder et al., 2016). A traditional teaching model lacks an assessment of one-to-one interaction and peer-feedback unlike PLTL. Peer leaders are expected to provide helpful hints and comments during students' discussion, but not reveal solutions, in contrast to traditional teaching models (Cracolice and Deming, 2001).

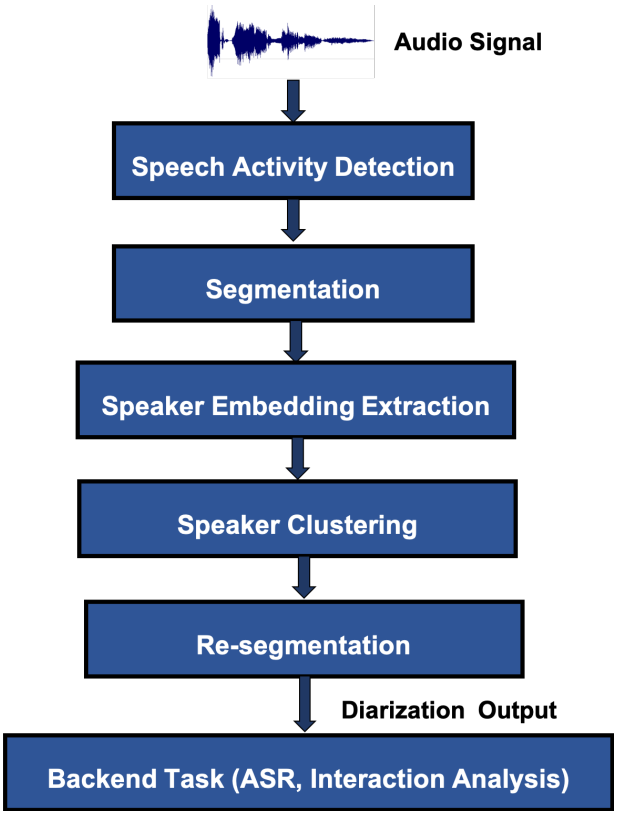


Figure 1.1. Systems view of speaker diarization pipeline for practical applications. It consists of four main steps: (i) speech activity detection (SAD), (ii) segmentation, (iii) speaker embedding extraction, (iv) speaker clustering, and (v) re-segmentation. The diarization output is used for backend tasks such as automatic speech recognition (ASR), interaction analysis etc.

Fig. 1.1 summarizes the core components of a general speaker diarization system used in several practical applications. Speaker diarization is an unsupervised/semi-supervised task, which consists of four sub-systems: (i) speech activity detection (SAD), (ii) segmentation and speaker embedding extraction, (iii) speaker clustering; and (iv) frame-level re-segmentation which can be optional. In earlier studies, researchers have found that each of these sub-systems can be studied independently (Sinclair and King, 2013). Even after several years of research on diarization techniques, state-of-the-art methods do not perform well on naturalistic audio streams. This dissertation leverages recent advancements in deep learning and model-based clustering for improving speaker diarization applied to naturalistic audio streams. We also

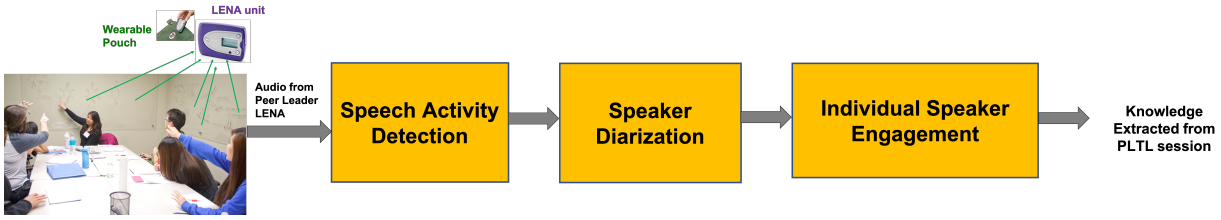


Figure 1.2. Overall view of dissertation scope. The primary focus is on improving SAD and speaker diarization. Secondary emphasis is on knowledge extraction and interaction analysis for small-group (6-10 participant) conversations. For the secondary goal, several approaches are investigated for extracting behavioral characteristic in PLTL sessions.

study speaker recognition based on training advanced speaker embedding (model) extractors. This dissertation presents research advancements on all sub-systems within the speaker diarization pipeline.

1.2 Dissertation Scope

Most state-of-the-art diarization techniques aim to address the two-speaker diarization problem (e.g., 2-person telephone conversations, broadcast news interviews). These are more structured and significantly simpler as compared to collaborative small-group discussions with 8-10 individuals. Furthermore, most state-of-the-art techniques are developed for telephone speech which is both cleaner/noise-free and a simpler conversation structure for speech processing versus free-form multi-speaker PLTL-type naturalistic audio. This dissertation primarily focuses on improving speaker diarization for naturalistic audio streams. Specifically, we investigate SAD, speaker recognition, and speaker modeling with deep neural network techniques, and speaker clustering. A secondary level focus is on knowledge extraction for interaction analysis of small-group conversations. Interaction analysis of PLTL session is a target application for knowledge extraction. Fig. 1.2 presents a high-level overview of the dissertation scope, based on multi-speaker PLTL conversational interactions, multi-stream audio capture, SAD, speaker clustering and recognition, followed by probe analysis of students engagement.

Given the unavailability of annotated data with speech/non-speech labels for general diarization tasks, we propose to formulate effective unsupervised SAD technique using novel features and three decisions backends. We model the raw waveform using a recently developed SincNet (Ravanelli and Bengio, 2018) convolutional neural architecture. We improve SincNet by proposing discriminative loss functions based on Center Loss (CL) and additive margin (AM)-softmax. We replace the Softmax loss function in the standard SincNet by our AM-Softmax to obtain AM-SincNet; also, the joint CL and Softmax to obtain our CL-SincNet, and the joint CL and AM-Softmax to obtain AM-CL-Softmax. These SincNet models are trained using TIMIT and Librispeech corpora. We also employ supervised transfer learning (STL) where the SincNet is trained first on TIMIT, and later re-trained on Librispeech. These resulting SincNet models are trained for frame-level speaker recognition. Once trained, we leverage these models for speaker embeddings extraction for speaker diarization. We propose novel neural architecture for speaker recognition and diarization. After considering the two tasks of SAD and speaker recognition, we move on to investigate three model-based approaches for speaker clustering: (i) mixture of von Mises-Fisher distributions (movMF); (ii) Normalized Fuzzy C-means (NFCM); and (iii) Toeplitz Inverse Covariance (TIC). This completes the intended dissertation scope for advancing diarization. A secondary level focus is also considered for knowledge extraction after speaker diarization is completed. Here, we consider speech features for a probe investigation of individual speaker engagement. We leverage the CRSS Speaker Profiler system for knowledge extraction using the following four speech/speaker attributes: (1) physical task stress, (2) whisper detection, (3) emotion recognition, and (4) Lombard effect. This completes the scope for secondary level probe investigation.

1.3 Dissertation Contributions

The main objective of this dissertation is to advance speaker diarization for naturalistic audio streams. In addition, we also improve speaker recognition using novel variations of the SincNet architecture and discriminative loss functions. This study included a formal data collection of CRSS-PLTL corpora that contains recordings of small-group student learning conversations in naturalistic scenarios. The goal of this dissertation it to design new algorithms for improving each sub-system in diarization pipeline. The specific contributions of this dissertations are as follows:

1. Frequency-Dependent Kernel (FDK) features for Robust SAD

   We propose FDK features as a novel way of decomposing the speech signal such that distinct frequency-dependent kernels are used for analyzing different frequency bins. We employ frequency-dependent Gaussian kernels where the width of each kernel is inversely proportional to frequency bin. In this manner, we have narrow kernels are available for smaller frequency bins and wider ones for higher frequency bins. FDK features aim to provide a generalized decomposition of the speech energies across time-frequency locations. Here, eight statistical descriptors are derived from the logarithm of the absolute value of the FDK feature vector corresponding to each frame. These statistical descriptors are mean and variance normalized and later processed with principal component analysis (PCA). The first principal component is chosen as the final FDK-SAD feature. This feature is leveraged with three proposed decisions backends for achieving unsupervised/semi-supervised SAD.

2. SAD Decision Backends: (i) VMGMM; (ii) DipSAD, and (iii) D-SAD

   Three decision backends are proposed for SAD: (i) Variable Model Size Gaussian Mixture Model (VMGMM); (ii) Hartigan Dip test for robust feature clustering (DipSAD), (iii) Density-SAD (D-SAD) which fits a linear curve for the cumulative distribution of

SAD features to derive an overall decision threshold. We combine these three decision backends with the FDK-SAD feature to obtain three unsupervised/semi-supervised SAD systems. Comparative evaluation experiments are used to highlight the competitive strength of the proposed SAD techniques over current state-of-the-art approaches.

3. Speaker Modeling:(i)SincNet,(ii)AM-SincNet,(iii)CL-SincNet, and(iv)AM-CL-SincNet

   Raw waveform modeling with a SincNet convolutional neural network is used to develop an advanced speaker modeling structure. This architecture is trained for frame-level (10ms) speaker recognition. We incorporate discriminative loss functions, additive margin (AM)-Softmax, and Center Loss (CL), to formulate advanced SincNets which are used for speaker recognition. These SincNet models are trained using out-of-domain data such as TIMIT and Librispeech corpora. The trained SincNet is adopted for unsupervised vanilla transfer learning (VTL) in order to extract frame-level speaker embeddings from in-domain CRSS-PLTL and AMI data. Experiments are performed with supervised transfer learning (STL) for data efficient training of SincNet. STL approach first uses TIMIT where the output layer is discarded and two new layers are added for training on new unseen Librispeech corpus. The hyper-parameters are optimized, with subsequent discussion to highlight the importance in achieving robust diarization performance. SincNet and its variants extract speaker embeddings from short speech frames of 100-200ms with a subsequent 10ms skip rate between frames. This approach eliminates any need for speaker change detection. SincNet embeddings are found to be superior than i-Vectors, since i-Vectors do not perform well for short duration utterances. The proposed novel SincNet architectures also converge faster that traditional SincNet. Neural speaker modeling using SincNet architecture performs significantly better than an i-Vector baseline. Initial work also considers unsupervised denoising autoencoder (DAE) for a meeting-specific speaker embedding extractor and an HMM for joint segmentation and speaker clustering.

4. Speaker Clustering: (i) movMF, (ii) NFCM, and (iii) TIC

   In this area, three model-based approaches are proposed for speaker clustering. A mixture of von Mises-Fisher distributions (movMF) is proposed for length-normalized speaker embeddings from a meeting recording. In this case, each component in mixture model represents one speaker. Standard expectation maximization (EM) is used for iterative speaker clustering which alternates between cluster assignment and re-estimation of the movMF model parameters. The second approach is based on a normalized Fuzzy C-means (NFCM) speaker clustering solution which is suitable for length-normalized speaker embedding. This leverages recent developments on Fuzzy C-means for soft-clustering using length-normalized data. Soft speaker clustering provides the possibility of a more flexible decision making process. The third approach attempts to learn a Markov Random Field (MRF) correlation network for each speaker. This method quantifies each speaker using a Toeplitz Inverse Covariance matrix (TIC), hence the name TIC speaker clustering. It is based on a dynamic programming (DP) strategy as well as optimizing a Toeplitz graphical lasso optimization problem. A set of comparison experiments is performed for different combinations of speaker embeddings and clustering approaches. It is noted that a cosine K-means approach is adopted as the baseline for speaker clustering.

5. Knowledge Extraction and Interaction Analysis

   This represents a secondary probe focus which begins with the diarization output, with probe analysis to extract knowledge relating to behavioral metrics for small-group PLTL conversations. We consider several metrics to assess subject engagement in multi-speaker PLTL learning spaces.

In addition to these algorithmic advancements, we established the CRSS-PLTL corpus for audio-based analysis of PLTL sessions (Dubey et al., 2016). Significant effort is dedicated

in effective corpora development, annotation and data preparation for conducting research studies reported in this dissertation.

1.4 Dissertation Outline

This dissertation is organized into seven chapters which are described as follows. A separate list of abbreviations, figures and tables are included before Chapter 1, to help in navigating this dissertation.

- Chapter 2, Background and Tools:

  In Chapter 2, we provide relevant background material in understanding the later chapters of this dissertation. First, Peer-Led Team Learning (PLTL) is introduced and the need for speech-based interaction analysis in PLTL is summarized. We review the literature in speech activity detection (SAD) and speaker diarization. Secondly, we establish the CRSS corpora used in this research study. We utilized CRSS-PLTL and AMI corpora for speaker diarization. Thirdly, we present CRSS-LDNN corpora that contains recordings of multi-layer noise in naturalistic scenarios. Along with CRSS corpora, we also summarize standard public corpora such as NIST-OpenSAD-2015, NIST-OpenSAT2017 and RedDots corpora. This chapter ends by defining the evaluation metrics for SAD and speaker diarization.

- Chapter 3, Speech Activity Detection:

  In Chapter 3, we present the proposed methods for SAD. Specifically, we propose novel frequency-dependent kernel (FDK) features that discriminate speech from non-speech. Next, we propose three decision backends for generating SAD labels from an input feature stream, that include: (i) Variable Model-size Gaussian Mixture Model (VMGMM); (ii) DipSAD that is a robust feature clustering based on Hartigan Dip test; (iii) Density-SAD (D-SAD) that derives a decision threshold by fitting a straight

line to the cumulative feature distribution. We compare these proposed methods with several state-of-the-art SAD techniques. Our SAD evaluation includes CRSS corpora and standard public available corpora. This chapter ends with a discussion results obtained with highlights of advancements for SAD in naturalistic audio streams.

- Chapter 4, SincNets based Speaker Recognition and Diarization:

  In Chapter 4, we start by covering state-of-the-art speaker embeddings from deep neural networks (DNNs). We propose vanilla transfer learning (VTL) based on SincNet for extracting speaker embedding. SincNet is convolutional DNN architecture for efficient modeling of raw waveform speech. We also proposed Center Loss (CL) for SincNet, and combine center loss with softmax and additive margin softmax (AM-Softmax) to obtain variants of SincNets namely, AM-SincNet, CL-SincNet, AM-CL-SincNet. CL-SincNet and AM-CL-SincNet for speaker recognition. We discuss results on CRSS-PLTL and AMI corpora on speaker diarization task and TIMIT and Librispeech data for speaker recognition. We further leverage transfer learning (TL) for efficient training of SincNets using multiple datasets.

- Chapter 5, Robust Speaker Clustering :

  In Chapter 5, we describe the proposed methods for speaker clustering which is the most important component in speaker diarization. First, we present past state-of-the-art methods followed by three new proposed approaches: (i) mixture of von Mises-Fisher distributions (movMF), (iii) Normalized Fuzzy C-means (NFCM), and (iii) Toeplitz Inverse Covariance (TIC) speaker clustering. We perform experiments to benchmark these methods over naturalistic corpora such as CRSS-PLTL and AMI meeting corpus. We report Diarization Error Rate (DER) % performance for all experiments.

- Chapter 6, Knowledge Extraction for PLTL Interaction Analysis:

  In Chapter 6, we present a probe study for extracting knowledge metrics related to

small-group conversations. This section builds on availability of diarization output from PLTL recordings. Specifically, we consider three components namely: (i) Unsupervised dominance score (DS); (ii) Pitch-based approach for detection of Question inflections; (iii) Energy and pitch-based approach for emphasis detection. Next, we present an idea for measuring engagement based on a separately developed multi-speaker style CRSS Speaker Profiler. The Speaker Profiler system detects four speaker-style attributes namely, (1) Emotion; (ii) Lombard effect; (iii) Physical Task stress; and (iv) Whisper.

- Chapter 7, Summary and Conclusions:

  Finally, in Chapter 7 we summarize the research and highlight improvements over current state-of-the-art solutions. We draw conclusions based on experimental results, and explain reasons for effectiveness and robustness of the proposed algorithms. We close this chapter by pointing towards directions for future work stemming from research contributions made by this dissertation.

CHAPTER 2

BACKGROUND AND TOOLS

In this chapter, we describe the background materials and technological tools that are needed to understand the research proposed in this dissertation. Background material is intended to augment the reader's understanding of underlying problems that are solved by algorithms discussed in later chapters.

2.1 Peer-Led Team Learning

Peer-led team learning (PLTL) is an established teaching paradigm for undergraduate STEM courses implemented in US universities (Tien et al., 2002; Snyder and Wiles, 2015). It is popular paradigm in undergraduate courses at many US universities and gaining attention in other countries as well. PLTL model is extensively studied by education researchers who found it augments the student's classroom studies (Cracolice and Deming, 2001; Carlson et al., 2016). Each team is assigned a peer leader who coordinate discussions among students, and facilitate collaborative problem solving. The peer leaders have passed the same course in earlier semester and thus they are aware of the challenges in learning the subject (Wamser, 2006). Peer leader knew the strategies that could help in mastering the technical content of the course. Peer leaders are not supposed to tell the solutions, rather they provide helpful hints and direction that could guide the students to collaboratively solve the problems (Lyle and Robinson, 2003; Roh et al., 2016).

2.2 CRSS-PLTL and CRSS-PLTL-II Corpus

This section briefly describes the CRSS-PLTL corpora that motivated the research discussed in this dissertation. We established CRSS-PLTL and CRSS-PLTL-II corpora that contains naturalistic interactions between 8-10 speakers. In association with the UT-Dallas Student

Figure 2.1. PLTL scenario for data recording in Student Success Center at The University of Texas at Dallas.

Success Center, we collected two corpora namely CRSS-PLTL and CRSS-PLTL-II for assessment of speech communication in PLTL sessions (Dubey et al., 2017). During PLTL sessions, each participant wore a LENA device (with not-so-close microphone) for collecting naturalistic audio (Hansen et al., 2018). Each student wore a wearable pouch containing LENA digital recorder as shown in Fig. 2.1. LENA device could record audio signals for long-duration of up to sixteen hours and had been used in a variety of human-to-human communication research, for example adult-child interaction (Sangwan et al., 2015) etc.

CRSS-PLTL contains multi-stream audio recordings from five PLTL teams from undergraduate Chemistry course over 11 week each, thus leading to 55 sessions. Similarly, CRSS-PLTL-II collected audio recordings of five teams chosen from undergraduate Calculus-II course leading to 55 sessions. Each PLTL session lasts for approximately 80 minute and constitute discussions between 6-8 students plus a peer-leader. Peer leader guides the group to arrive at correct solutions without explicitly telling the solution. Each of these corpora had approximately 300+ hours of audio, data from weekly PLTL sessions. Short utterances and rapid turn-taking were salient features of PLTL discussions. In this manner, we collected multi-stream audio for each session (number of streams was same as total participants). The salient features of this data are: (i) many segments with overlapped-speech; (ii) short conversational-turns; (iii) multiple noise-sources; and (iv) significant reverberation. These factors made PLTL speaker diarization challenging.

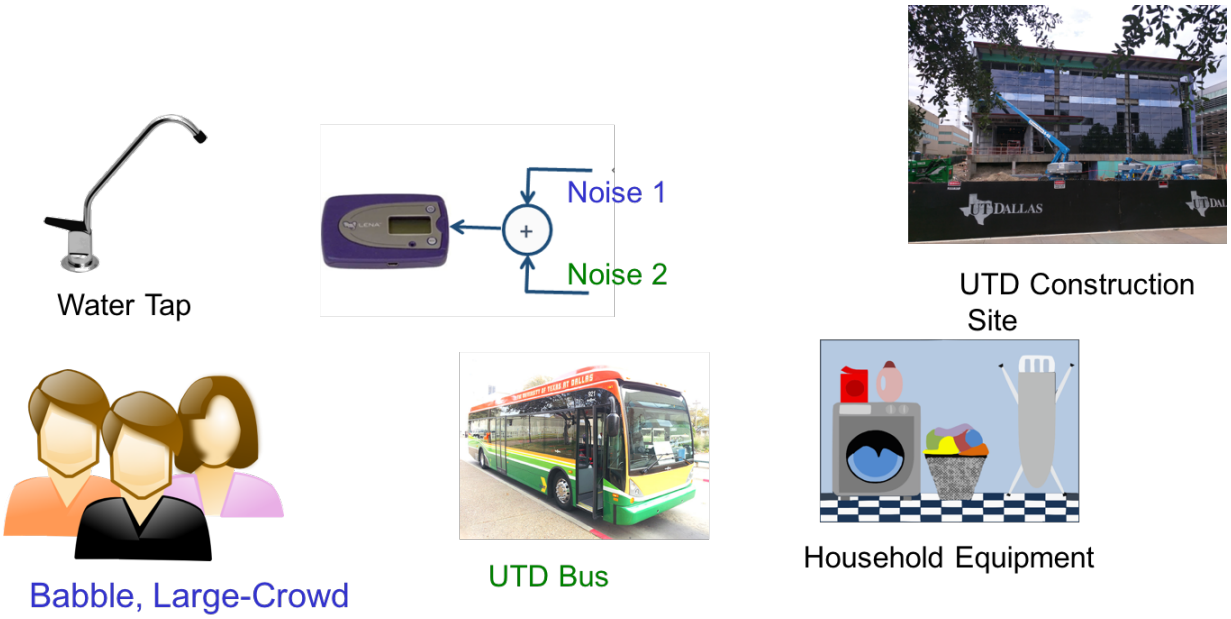


Figure 2.2. Scenarios where CRSS-LDNN noise corpus was recorded. Most situations has two or more noise sources active simultaneously.

2.3 Multi-layer Noise

Multi-layer noise refers to scenarios where multiple noise-sources are simultaneously active. For instance, many situations in daily life can result into mixing of any two or more of these noise types, namely, broadband (white Gaussian noise) is stationary noise, tonal/periodic (harmonic noise), and impulsive noise that belong to non-stationary types. Multi-layer noise referred to simultaneous presence of two or more noise-sources where each of such noises could exist alone as well. The babble noise and bus-engine noise present along with occasional impulsive-noise over long-duration is an example of such scenario.

2.4 CRSS Long-Duration Naturalistic Noise Corpus

Continuing from our discussion on multi-layer noise in Sec. 2.3, we present the CRSS long-duration naturalistic noise (CRSS-LDNN) corpus in this section. This corpus was collected using wearable LENA units at 16 kHz sampling rate with 16 bit precision in *.wav* format.

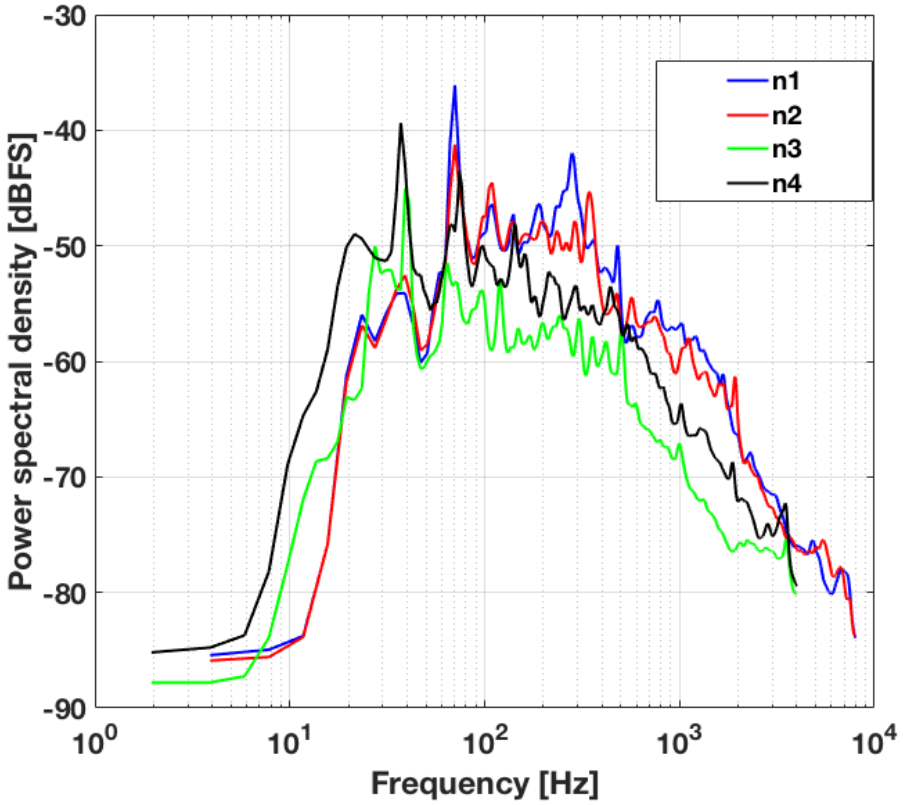


Figure 2.3. Long-term spectrum of four noise samples of duration more than 20 minutes chosen from CRSS-LDNN corpus. We can see the variety of prominent frequencies in these four noise samples n_1, n_2, n_3 and n_4.

This data would be released to speech community (http://crss.utdallas.edu). During the summer semester, a CRSS student wore a LENA device that was switched ON when multiple noise-sources were present. Fig. 2.2 shows the scenarios where CRSS-LDNN noise corpus was recorded. Most situations has two or more noise sources active simultaneously. In this way, the data was collected in naturalistic scenarios with uncontrolled mixing of various noise-sources. This corpus was supposed to provide naturalistic multi-layer noise recordings for evaluation of robust speech algorithms. We used the CRSS-LDNN data for corrupting the PLTL evaluation set for standalone SAD evaluations in this chapter. This corpus consisted of approximately 19 hours of noise data. The CRSS-LDNN noise data was more challenging as compared to existing noise corpora such as NOISEX (Varga and Steeneken, 1993) containing single

noise recordings. We collected a naturalistic noise corpus named the CRSS long-duration naturalistic noise (CRSS-LDNN) corpus. It contains noise data collected using wearable LENA units (Sangwan et al., 2015). The diversity in noise-sources includes construction noise, multi-speaker babble, large-crowd noise, vehicle/bus noise on the road, home environment noise etc. Fig. 2.3 shows the long-term spectrum of four noise samples chosen from CRSS-LDNN corpus. Each of these noise samples has duration of more than 20 minutes. We can clearly see the variety of prominent frequencies in these four noise samples. We would revisit CRSS-LDNN corpus while evaluating proposed SAD algorithms in Chapter 3.

2.5 Zero-Resource Speech Processing

Zero-resource speech processing refereed to systems with almost zero linguistic resources. It deals with unsupervised discovery of linguistic units from raw speech in an unknown language (Versteegh et al., 2015). state-of-the-art speech systems were trained on massive datasets with human annotations. However, such supervised methods would have language and/or channel mismatches when used for zero-resource speech applications where manually annotated data is either scarce or unavailable. Zero-resource speech processing explore systems that could be developed for a new language starting from scratch. It rely on robust unsupervised SAD for efficient processing. Such paradigms were also applicable for technologies involving under-resourced languages and/or dialects.

2.6 Supervised SAD

Supervised speech activity detection approaches were machine learning systems trained on annotated audio data. Such methods either focused on finding better generative features such as bottleneck features or discriminative classifiers such as deep neural networks (Zhang and Wu, 2013). Many SAD algorithms were developed as part of the DARPA RATS (Walker and Strassel, 2012a) program (Ng et al., 2012; Saon et al., 2013; Thomas et al., 2015; Graciarena

et al., 2013; Novotney et al., 2016; Karakos et al., 2016). For example, one team proposed combining several features for supervised SAD for DARPA RATS (Graciarena et al., 2013). Specifically Mel-frequency cepstral coefficients (MFCC), Gabor features processed with multilayer perceptron (MLP), Combo-SAD features, sub-band autocorrelation (SAcC) with MLP post-processing, and multi-band comb-filter F0 (MBCombF0) voicing were combined. This combining procedure led to significant gains in SAD accuracy (Graciarena et al., 2013). A solution for DARPA RATS phase 2 evaluation (Saon et al., 2013) consisted of multi-pass HMM segmentation and combined features for training feed-forward and convolutional neural networks.

Joint use of source and filter-based features was leveraged for supervised SAD (Drugman et al., 2016). Several feature sets were used for training neural network on multi-conditioned TIMIT data (Garofolo, 1993). The source and filter information were merged at the feature and score level out of which the score fusion performed better (Drugman et al., 2016). A maximum-margin clustering approach based on support vector machines (SVMs) was adopted for unsupervised SAD (Wu and Zhang, 2011). Two features, namely multiple observation signal-to-noise-ratio and multiple observation maximum-probability were proposed for maximum-margin clustering (Wu and Zhang, 2011). The multiple observation likelihood ratio test (MO-LRT) was used for robust SAD under noisy conditions. It out-performed the single observation likelihood ratio test (SO-LRT) that required an empirically tuned hangover scheme (Ramírez et al., 2005). MO-LRT leveraged long-term information for deriving an optimal decision rule (Ramírez et al., 2005).

A SAD system based on Gaussian mixture models (GMMs) and multi-layer perceptron was developed for the DARPA RATS program (Ng et al., 2012). This system leveraged a robust front-end, feature normalization, dimensionality reduction and score normalization (Ng et al., 2012). Researchers proposed a two-stage SAD based on an explicit model of phonetic information (Ferrer et al., 2016). The first step consisted of training a bottleneck deep neural

network (DNN) for predicting the senone posteriors. In the next step, activations of the bottleneck layer were used for training another DNN for predicting speech and non-speech posteriors. Though the proposed system led to significant improvements over a baseline single DNN system under matched conditions, it failed to provide significant gains for mismatched channels (Ferrer et al., 2016). The improvements in IBM SAD system for DARPA RATS involved joint training of convolutional (CNN) and feed-forward DNNs with temporal and spectral features. Improved CNN-DNN model led to significant gains in SAD accuracy under matched conditions (Thomas et al., 2015).

Researchers explored fusing six SAD systems including two supervised and four unsupervised for the NIST-OpenSAD-2015 data (Kinnunen et al., 2016). This study concluded that the channel detection improved the performance on development set but failed to generalize further (Kinnunen et al., 2016). i-Vectors are established approach for speaker and language recognition. These were recently used for segment-level SAD derived from the generalized likelihood ratio (GLR), Bayesian information criterion (BIC), K-means and GMM clustering (Khoury and Garland, 2016). This segment-level i-Vector SAD was found to be more accurate than a frame-level GMM baseline on the NIST-OpenSAD-2015 data (Khoury and Garland, 2016). The SRI NIST-OpenSAD system utilized three different development sets derived from the provided corpus (Graciarena et al., 2016). The fusion of acoustic, voicing and bottleneck features was used for unsupervised test-adaptive calibration. The feature normalization had a significant impact on SAD accuracy (Graciarena et al., 2016). The BBN OpenSAD system employed supervised, unsupervised and active learning-based model adaptation for SAD over unseen channels (Karakos et al., 2016; Novotney et al., 2016). The long short-term memory (LSTM) neural network SAD models were adapted for reducing the variability between training and testing data. Unsupervised adaptation used SAD labels automatically generated by a baseline model. Limited amounts of human annotations from unseen channels was utilized for supervised model adaptation (Karakos et al., 2016; Novotney

et al., 2016). Researchers further considered active learning-based supervised adaptation where the annotations were automatically selected for maximizing the performance (Karakos et al., 2016; Novotney et al., 2016). This approach leveraged output of SAD systems and led to significant gains in accuracy as compared to random selection of annotations (Karakos et al., 2016; Novotney et al., 2016).

2.7 Unsupervised SAD

Unsupervised SAD using energy based likelihood ratio tests was proposed in (Sohn et al., 1999). Unsupervised methods were designed to work in a variety of acoustic conditions and could sometimes better manage mis-matched conditions (Graciarena et al., 2016). In past studies such as (Ramırez et al., 2004; Ghosh et al., 2011) the long-term speech information was utilized for unsupervised SAD under noisy conditions. Recent work in (Sholokhov et al., 2018) summarized the SAD developments in context of semi-supervised and unsupervised techniques. It introduced the idea of semi-supervised learning in conventional expectation-maximization (EM) algorithm for GMMs (Sholokhov et al., 2018).

A hard-clustering approach using sub-band log-energy features was proposed for unsupervised SAD (Górriz et al., 2006). It performed better than the standard SAD systems such as ITU-T G.729, ETSI GSM AMR and ETSI AFE on a Spanish database. Another SAD approach used long-term temporal and spectral features in a statistical model for noise robustness (Fukuda et al., 2010). It led to improved performance as compared to ETSI AFE-SAD at low SNRs (Fukuda et al., 2010). A generalized Gamma distribution based likelihood ratio test was considered for SAD in (Shin et al., 2010). It outperformed SAD based on conventional Laplacian and Gamma distributions. A variational Bayes approach for SAD employed two parallel classifiers for noise-only and speech-with-noise (Cournapeau et al., 2010). Online expectation maximization was used for simultaneous adaptation of statistical model and decision threshold (Cournapeau et al., 2010).

2.8 NIST-OpenSAD-2015

NIST organized OpenSAD evaluation for promoting research in robust SAD for degraded military communication channels (NIST NIST, a). This data were derived from the DARPA RATS program (Walker and Strassel, 2012b). Six channels namely B, D, E, F, G, H and the clean source (src) channel from the DARPA RATS were included in the training set. The data consisted of telephonic conversations from source channel that were re-transmitted through these channels. Thus, it included clean and noisy audio recordings from three languages namely Levantine Arabic (alv), American English (eng) and Urdu (urd) (NIST NIST, a).

2.9 NIST-OpenSAT-2017 Public Safety Communications Corpus

NIST organized the OpenSAT pilot evaluation in spring last year. This evaluation targeted domains that were expected to be challenging for the current state-of-the-art. We chose the public safety communications (PSC) corpus from OpenSAT data for SAD evaluations. This data consisted of audio logs in English language. It contained audio data from sofa super store fire (SSSF) dispatcher that occurred on June 18, 2007 in Charleston, South Carolina. It constituted real fire-response operational data that could not be duplicated through controlled scientific collection (NIST NIST, b). Also, these audio recordings contained sensitive and disturbing content such as pleas from the trapped fire fighters. This data were rich in naturalistic distortions such as (i) land mobile radio transmission effects; (ii) speech under cognitive and physical stress; (iii) varying background-noise types and levels (NIST NIST, b). It had six audio recordings each of approximately five-minute duration, thus making up a total 30 minutes of `dev` data. It was provided as 16-bit signed integer PCM at 8 kHz sampling rate.

2.10 Speaker Diarization

Speaker Diarization answer *who spoke and when?* in an audio stream (Yella and Stolcke, 2015; Tranter and Reynolds, 2006). It usually consists of several components such as speech activity detection (SAD), initial-segmentation and speaker modeling, speaker clustering, and re-segmentation (Tranter and Reynolds, 2006; Anguera et al., 2012). Automatic interaction analysis in PLTL sessions would help education researchers to obtain insights into how learning outcomes are impacted by individual participation, group behavior, and team dynamics. Speaker diarization front-end is used for audio-based interaction analysis in PLTL sessions. The challenges in speaker diarization is application-dependent. Domains involving practical application of speaker diarization are understanding and transcription of broadcast news, audio-recorded meetings, telephonic conversations (Huijbregts and van Leeuwen, 2012) etc. NIST Rich Transcription evaluations focused on broadcast-news and meetings audio while NIST SRE evaluations had summed-channel telephone data (Anguera et al., 2012). Speaker Diarization for naturalistic interactions such as Peer-Led Team Learning (PLTL) sessions is a challenging task. Diarization involve extracting i-Vectors/speaker features from short speech-segments (typically one-second) unlike speaker verification where complete-utterance is used for extracting i-Vectors.

Researchers proposed a scheme for joint segmentation and clustering for diarization (Anguera et al., 2012). Recently, researchers combined audio and visual cues in spectro-temporal fusion for diarization (Gebru et al., 2018). This approach is suitable for scenarios that has video recordings of spontaneous interactions among several speakers. Practical applications of speaker diarization (Dubey et al., 2016a) include broadcast new analysis, low-latency speaker spotting (Patino et al., 2018) and behavioral study (Dubey et al., 2017). Given the importance of robust clustering for speaker diarization, several approaches were developed such as agglomerative hierarchical clustering (AHC) (Sun et al., 2010), top-down clustering (Meignier et al., 2006), cosine K-means clustering, and HMM-based speaker clustering (Ajmera and Wooters,

2003) etc. In (Zhu et al., 2005), the MAP-adapted Gaussian mixture-models (GMMs) were combined with Bayesian information criterion (BIC) for speaker diarization. A reduced complexity clustering approach leverages modified integer linear programming (ILP) (Dupuy et al., 2014). Recently, speaker diarization based on i-Vectors probabilistic linear discriminant analysis (PLDA) approach was analyzed in details (Salmun et al., 2017). Weighted GMMs were utilized for multi-speaker segmentation for DARPA Hub4 Broadcast News 1997 evaluation (Huang and Hansen, 2006). Unsupervised calibration of PLDA scores was used within i-Vector clustering framework for CALLHOME corpus (Sell and Garcia-Romero, 2014).

2.10.1 i-Vector Speaker Model

Diarization involve extracting i-Vectors from short speech-segments (typically one-second) unlike speaker verification where complete-utterance is used for extracting i-Vectors. Numerous techniques were developed for i-Vector clustering based on cosine similarity (Senoussaoui et al., 2014; Castaldo et al., 2008). The i-Vector framework combines the speaker and channel variability sub-spaces of linear distortion model into a total-variability space represented by matrix $\mathbf{T}$ (Dehak et al., 2011; Hansen and Hasan, 2015). A speaker-and-session-dependent GMM super-vector, $\mathbf{S}$ is decomposed as

$$\mathbf{S} = \mathbf{S_{ubm}} + \mathbf{Tw}, \tag{2.1}$$

where $\mathbf{S_{ubm}}$ is the Universal Background Model (UBM) super-vector (Dehak et al., 2011). The latent variables, $\mathbf{w}$ are i-Vectors. The total-variability matrix $\mathbf{T}$ is a low-rank projection matrix that maps high-dimensional speaker super-vectors to low-dimensional total-variability space (Dehak et al., 2011; Hansen and Hasan, 2015). We use frame-level 20-MFCC features extracted from 40ms windows at 10ms skip-rate. A UBM with 512 components is trained for i-Vector extraction (Dehak et al., 2011). Given the short speaker-turns in PLTL, we choose the i-Vector dimension as 75. We post-processed the segment-level i-Vectors with PCA for

Figure 2.4. Scenario for data recording in AMI meeting corpus with four speaker discussing a project.

dimensionality reduction followed by length-normalization (Garcia-Romero and Espy-Wilson, 2011).

2.10.2 Baseline Diarization System

In this dissertation, we chose a speaker diarization baseline that consists of i-Vectors and Cosine K-means clustering (Zhong, 2005). The cosine similarity was previously used for comparing length-normalized i-Vectors in K-means and mean-shift clustering paradigms (Dehak et al., 2011; Castaldo et al., 2008; Senoussaoui et al., 2014). Cosine K-means projects the estimated cluster-centroids onto the unit hypersphere at the end of each maximization-step unlike the conventional K-means. It is a widely used approach for i-Vector based speaker clustering.

2.11 AMI Corpus

Augmented Multi-party Interaction (AMI) corpus provides speaker annotated multi-modal data from meeting scenarios (see Fig. 2.4). The audio data was provided with reference speaker annotations. We choose 6 meetings from AMI corpus as evaluation set for speaker

Table 2.1. Details of 12 meeting subset chosen from AMI corpus (McCowan et al., 2005). Duration is rounded off to next integer value.

Meeting ID	**Audio Duration (min)**	**Speaker Count**
IS1000a	27	4
IS1001a	16	4
IS1001b	36	4
IS1001c	25	4
IS1003b	28	4
IS1003d	36	4
IS1006b	37	4
IS1006d	31	4
IS1008a	16	4
IS1008b	30	4
IS1008c	26	4
IS1008d	25	4

diarization experiments reported in this dissertation. The audio duration and number of speakers in 12 meetings set of AMI is shown in Table 2.1. We used *mixed headset audio* for experiments reported in this dissertation. Our AMI evaluation set consists of sessions: IS1006d (31 min.), IS1003d (36 min.), IS1001a (16 min.), IS1000a (27 min.), IS1003b (27 min.) and IS1008d (25 min.). Each of the three meetings has four speakers discussing a project such as design of a new remote control device.

2.12 RedDots Corpus for Text-Dependent Speaker Verification

A smartphone app was used for crowd-sourcing the RedDots data collection. The audio was recorded by the individuals using different handsets under variable acoustic conditions. Native as well as non-native English speakers with diverse accents participated in data collection across the globe. We chose the Q4 release of RedDots corpus (Lee et al., 2015) for text-dependent speaker verification experiments reported in this chapter. We selected part 1 (fixed pass-phrase task) male subset from the corpus. In this subset, the speakers pronounce a set of fixed pass-phrases identical for all speakers. The male subset of part 1 was used as it had more number of trails than the female subset.

2.13 LibriSpeech

LibriSpeech is a open-source corpus of approximately 1000 hours read English speech sampled at 16kHz (Panayotov et al., 2015). This data is derived from read audiobooks from the LibriVox project, and has been carefully segmented and aligned (Panayotov et al., 2015). It is an ASR corpus extracted from public domain audio books from the LibriVox project. Recently, it was used in speaker recognition studies (Ravanelli and Bengio, 2018). We leveraged this corpus as out-of-domain data for training speaker recognition models.

2.14 Evaluation Metrics

In this section, we describe three evaluation metrics: (i) Detection Cost Function (DCF) for SAD; and (ii) Diarization Error Rate (DER) for diarization; and (iii) Mutual Information (MI) for diarization.

2.14.1 SAD Detection Cost Function (DCF)

NIST DCF is a metric used for benchmarking the SAD systems. It is defined as:

$$\text{DCF} = w * P_{fa} + (1 - w) * P_{miss} \tag{2.2}$$

where P_{fa} and P_{miss} are probabilities of false alarm and miss rate respectively, and w is a weight chosen from interval [0,1]. Since the focus of this dissertation is on PLTL speaker diarization where false alarms and miss rate are equally important we chose, w=0.5 for SAD experiments involving PLTL data.

2.14.2 Diarization Error Rate (DER)

Diarization error rate (DER) was used for scoring the systems with respect to ground-truth annotations. It was introduced in the NIST Rich Transcription Spring 2003 evaluation

(RT-03S). It is defined as the total percentage of reference time that is not correctly attributed to a speaker. Mathematically, DER is given as:

$$\text{DER} = \frac{\Phi_{fa} + \Phi_{miss} + \Phi_{spk}}{\Phi_{total}}, \tag{2.3}$$

where Φ_{total} is the total time of all reference segments, Φ_{fa} is the system speaker-time not attributed to the reference speaker, Φ_{miss} is the total reference speaker-time not attributed to a system speaker, and Φ_{spk} is the total reference speaker-time attributed to a wrong speaker.

2.14.3 Mutual Information (MI)

Similar to DER, frame-level mutual information (MI) is used for scoring the system-output with respect to reference speaker segmentation. MI quantifies the statistical-similarity between frame-level system-output and ground-truth. First of all, both ground-truth and system-output are converted to 10ms frame-level labels. Then, the frame-level MI (in bits) between system-output and ground-truth is mathematically defined as:

$$\text{MI} = \sum_{i=1}^{R} \sum_{j=1}^{S} \frac{n_{ij}}{N} \log_2 \frac{n_{ij} N}{r_i s_j}, \tag{2.4}$$

where R, S are the number of reference and system clusters, respectively; n_{ij} is the number of frames assigned to i-th reference and j-th system cluster; r_i, s_j are the number of frames assigned to i-th reference, and j-th system cluster, respectively; and N is the total number of frames. We compute MI values using the scoring scripts from *First DIHARD Challenge Evaluation* (Ryant et al., 2018).

2.15 Summary

This chapter reviews the background material needed to disseminate the research discussed in this dissertation. It can serve as a reference for tools used later in this dissertation. We covered the PLTL paradigm, CRSS corpora, standard tasks of SAD and speaker diarization.

In the end, we presented the evaluation metric for benchmarking the SAD and speaker diarization algorithms.

CHAPTER 3

SPEECH ACTIVITY DETECTION [1]

3.1 Introduction

Speech activity detection (SAD) is a key front-end in most speech systems (e.g., speaker verification, speech recognition, speaker diarization). Speech activity detection (SAD) systems discriminate between speech and non-speech segments within an input stream. Interest in robust SAD over degraded channels have existed for several years (Ramírez et al., 2005; Zhang and Wu, 2013; Shin et al., 2010; Sadjadi and Hansen, 2013). Two broad classes of SAD algorithms are: (1) supervised; (2) unsupervised. While supervised techniques require training on massive amounts of annotated data, unsupervised approaches do not require training on labeled data (Sohn et al., 1999). Supervised methods perform poorly on mismatched train and test conditions (Sholokhov et al., 2018; Sohn et al., 1999; Ramírez et al., 2005; Ferrer et al., 2016). In one major area, the DARPA RATS program supported SAD research in multiple phases leading to advanced developments (Ng et al., 2012; Saon et al., 2013; Thomas et al., 2015; Graciarena et al., 2013; Sadjadi and Hansen, 2013; Novotney et al., 2016). Despite several decades of research efforts, unsupervised SAD remains a challenging task for adverse acoustic conditions. Supervised SAD typically leverages acoustics models, and more recently machine learning models, trained on annotated data. For applications such as zero-resource speech processing and the NIST-OpenSAT-2017 public safety communications task, it might not be feasible to collect SAD annotations. Furthermore, SAD is challenging for naturalistic human engaging audio streams that contains multiple noise-sources that are active simultaneously.

[1] ©2018 IEEE. Portions Adapted, with permission, from H. Dubey, A. Sangwan, J. H. L. Hansen, "Leveraging Frequency-Dependent Kernel and DIP-Based Clustering for Robust Speech Activity Detection in Naturalistic Audio Streams," IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2018 Nov;26(11):2056-71.

In this chapter, we propose novel frequency-dependent kernel (FDK) based SAD features. FDK provides an enhanced spectral decomposition from which several statistical descriptors are derived. These statistical descriptors are combined by principal component analysis into one-dimensional FDK-SAD features. For leveraging the FDK-SAD features for efficient SAD, we propose two decision backends: (i) Variable Model-size Gaussian mixture model (VMGMM) ; and (ii) Hartigan dip-based robust feature clustering (DipSAD). While VMGMM is a model based approach, the DipSAD is non-parametric. We use both backends for comparative evaluations in two phases: (1) standalone SAD performance; (2) effect of SAD on text-dependent speaker verification using RedDots data. The NIST-OpenSAD-2015, NIST-OpenSAT-2017 and CRSS-PLTL corpora are used for standalone SAD evaluations. We also perform comparative studies of the proposed approaches with multiple baselines including SohnSAD, rSAD, Semi-Supervised Gaussian Mixture Model (SSGMM) , and Gammatone spectrogram features.

The core contributions of this chapter are:

- propose a set of novel FDK-SAD features.
- propose two alternative decision backends for SAD, namely VMGMM and DipSAD.
- evaluate the proposed SAD systems for two tasks: (1) standalone SAD, and (2) effect of SAD on text-dependent speaker verification using RedDots data.

3.2 Speech Activity Detection

Speech activity detection (SAD) systems discriminate between speech and non-speech segments in an audio signal. The interest in robust SAD over degraded channels have existed for several years (Ramírez et al., 2005; Zhang and Wu, 2013; Shin et al., 2010; Górriz et al., 2006). Two broad classes of SAD algorithms are: (1)supervised; (2)unsupervised. While supervised techniques require training on massive amount of annotated data, the unsupervised

approaches do not require training on labeled data (Sohn et al., 1999). Supervised methods perform poorly on mismatched train and test conditions (Sholokhov et al., 2018; Sohn et al., 1999; Sarikaya and Hansen, 1998; Ziaei et al., 2014; Ramírez et al., 2005; Ferrer et al., 2016). DARPA RATS program supported the SAD research in multiple phases leading to advanced developments (Ng et al., 2012; Saon et al., 2013; Thomas et al., 2015; Graciarena et al., 2013; Novotney et al., 2016; Graciarena et al., 2016). Despite several decades of research efforts, unsupervised SAD remained challenging for adverse acoustic conditions.

Before describing the proposed solutions, we considered the scenarios that require unsupervised SAD. In light of applications such as zero-resource speech processing and low-resource languages, it is impractical to annotate massive audio data for training supervised systems. Such applications demand developments of robust unsupervised or semi-supervised techniques. NIST-OpenSAT-2017 public safety communications (PSC) task had limited development data (approx. 30 minutes) acquired from real-world sensitive scenario that could not be duplicated through controlled scientific collection unlike DARPA RATS. Thus, it is impractical to obtain matched training data for PSC scenarios. Zero-resource speech processing demands no linguistic resource that rules out the possibility to train or adapt a supervised SAD.

The remainder of this chapter is organized as follows. We highlight the motivation and background work on SAD under different naturalistic scenarios in Sec. 3.3. Sec. 3.4 presents baseline methods for our subsequent experiments. Sec. 3.5 outlines the algorithmic development of FDK-SAD features. Sec. 3.6 develops the VMGMM backend. Sec. 3.7 describes the robust clustering approach in the second backend based DipSAD solution. Sec. 3.9 specifies the experimental setup reported in this chapter. Finally, we discuss the results from conducted experiments in Sec. 3.10.1. Lastly, we summarize our findings while pointing out the strengths and weaknesses of the propose approaches in Sec. 3.11.

3.3 Motivation and Background

Before addressing the formulation of the proposed solutions, it is important to understand the scenarios where unsupervised SAD is the right choice. In light of applications such as zero-resource speech processing and low-resource languages, it is impractical to annotate massive audio data for training supervised systems. Such applications demand developments of robust unsupervised or semi-supervised techniques. The NIST-OpenSAT-2017 (NIST NIST, b) public safety communications (PSC) task contains limited development data (approx. 30 minutes) acquired from real-world sensitive scenarios that could not be duplicated through controlled scientific collections, unlike the DARPA RATS corpus (Walker and Strassel, 2012a). Thus, it is impractical to obtain matched training data for some real-world scenarios. Zero-resource speech processing demands no linguistic resource that rules out the possibility to train or adapt a supervised SAD.

It is important to note that the PLTL data has (see section 2.6 for a description of the Peer-Led Team Learning corpus developments) : (1) a not-so-close body worn microphone; and (2) small physical movement of students, such as moving to white board and writing something. These issues can also make SAD a challenging task for human interactive conversations. In addition, significant reverberation and noise corrupt the speech data further.

3.3.1 Multi-layer noise

Multi-layer noise refers to scenarios with multiple noise-sources are simultaneously active in an audio stream. For this study, we collected a naturalistic noise corpus named CRSS long-duration naturalistic noise (CRSS-LDNN) corpus. It contains noise data collected using wearable LENA units (Hansen et al., 2017, 2016). Diverse noise sources include outdoor building construction noise, multi-speaker babble noise, large-crowd noise, vehicle/bus noise on the road, home environment noise etc. More details are presented in section 2.6.

3.3.2 Zero-resource speech processing

Zero-resource speech processing refers to systems which require almost zero linguistic resources. It deals with unsupervised discovery of linguistic units from raw speech in an unknown language (Versteegh et al., 2015). Current state-of-the-art speech systems are trained on massive datasets with extensive human annotations. However, such supervised methods experience language and/or channel mismatch when used for zero-resource speech applications where manually annotated data is either scarce or unavailable. Zero-resource speech processing explores systems that could be developed for a new language starting from scratch. This relies on robust unsupervised SAD for efficient processing. Such paradigms are also applicable for technologies involving under-resourced languages and/or dialects.

3.3.3 Supervised SAD

Supervised speech activity detection approaches historically relied on acoustic models such as Hidden Markov Model (HMM) (Saon et al., 2013), Gaussian mixture model (GMM) (Sadjadi and Hansen, 2013) etc. More recently, machine learning systems have emerged that are trained on large annotated audio data. Such methods either focus on finding better generative features such as bottleneck features or discriminatively trained classifiers such as deep neural networks (DNNs) (Zhang and Wu, 2013). Many such algorithms were developed as part of the DARPA RATS (Walker and Strassel, 2012a) program (Ng et al., 2012; Saon et al., 2013; Thomas et al., 2015; Graciarena et al., 2013; Novotney et al., 2016; Karakos et al., 2016). For example, one team propose combining several features for supervised SAD for DARPA RATS (Graciarena et al., 2013). Specifically, Mel-frequency cepstral coefficients (MFCC), Gabor features processed with multilayer perceptron (MLP), Combo features (Sadjadi and Hansen, 2013), sub-band autocorrelation (SAcC) with MLP post-processing, and multi-band comb-filter F0 (MBCombF0) voicing were combined. This combining procedure led to significant gains in SAD accuracy (Graciarena et al., 2013). A

solution for DARPA RATS phase 2 evaluation (Saon et al., 2013) consisted of multi-pass Hidden Markov Model (HMM) segmentation and combined features for training feed-forward and convolutional neural network (CNN).

The joint use of source and filter-based features has also been leveraged for supervised SAD (Drugman et al., 2016). Several feature sets were also used for training neural network on multi-conditioned TIMIT data (Garofolo, 1993). The source and filter information were then merged at the feature and score level out of which the score fusion performed better (Drugman et al., 2016). A maximum-margin clustering approach based on support vector machine (SVM) was adopted for unsupervised SAD (Wu and Zhang, 2011). Two features, namely multiple observation signal to noise ratio (SNR) and multiple observation maximum-probability were propose for maximum-margin clustering (Wu and Zhang, 2011). The multiple observation likelihood ratio test (MO-LRT) was used for robust SAD under noisy conditions. It outperformed the single observation likelihood ratio test (SO-LRT) that require an empirically tuned hangover scheme (Ramírez et al., 2005). MO-LRT leveraged long-term information for deriving an optimal decision rule (Ramírez et al., 2005).

A SAD system based on Gaussian mixture models (GMMs) and multi-layer perceptron was also developed for the DARPA RATS program (Ng et al., 2012). This system leverages a robust front-end, feature normalization, dimensionality reduction and score normalization (Ng et al., 2012). In another study, a two-stage SAD based on an explicit model of phonetic information was proposed (Ferrer et al., 2016). The first step consisted of training a bottleneck deep neural network (DNN) for predicting the senone posteriors. In the next step, activations of the bottleneck layer were used for training another DNN for predicting speech and non-speech posteriors. Though the propose system led to significant improvements over a baseline single DNN system under matched conditions, it failed to provide significant gains for mismatched channels (Ferrer et al., 2016). The improvements in the IBM SAD system for DARPA RATS involved joint training of convolutional (CNN) and feed-forward DNNs with temporal and

spectral features. An improved CNN-DNN model led to significant gains in SAD accuracy under matched conditions (Thomas et al., 2015).

Another study explored fusing six SAD systems including two supervised and four unsupervised for the NIST-OpenSAD-2015 data (Kinnunen et al., 2016). This study concluded that channel detection improves performance on the development set, but failed to generalize further (Kinnunen et al., 2016). In speech modeling, i-Vectors have been an established approach for speaker and language recognition. These have been used for segment-level SAD derived from the generalized likelihood ratio (GLR) , Bayesian information criterion (BIC) , K-means and GMM clustering (Khoury and Garland, 2016). This segment-level i-Vector SAD was found to be more accurate than a frame-level GMM baseline on the NIST-OpenSAD-2015 data (Khoury and Garland, 2016). The SRI NIST-OpenSAD system utilized three different development sets derived from the provided corpus (Graciarena et al., 2016). The fusion of acoustic, voicing and bottleneck features was used for unsupervised test-adaptive calibration. In this case, feature normalization had a significant impact on SAD accuracy (Graciarena et al., 2016). The BBN OpenSAD system employed supervised, unsupervised and active learning-based model adaptation for SAD over unseen channels (Karakos et al., 2016; Novotney et al., 2016). Long short-term memory (LSTM) neural network SAD models have also been adapted for reducing the variability between training and testing data. Unsupervised adaptation used SAD labels automatically generated by a baseline model. Limited amounts of human annotations from unseen channels have been successfully utilized for supervised model adaptation (Karakos et al., 2016; Novotney et al., 2016). Further, an active learning-based supervised adaptation was considered where the annotations were automatically selected for maximizing the performance (Karakos et al., 2016; Novotney et al., 2016). This approach leveraged the output of SAD systems and led to significant gains in accuracy as compared to random selection of annotations (Karakos et al., 2016; Novotney et al., 2016).

3.3.4 Unsupervised SAD

Unsupervised SAD using energy based likelihood ratio tests was propose in (Sohn et al., 1999). Unsupervised methods are designed to work in a variety of acoustic conditions and could sometimes better manage mis-matched conditions (Graciarena et al., 2016). In previous studies such as (Ramırez et al., 2004; Ghosh et al., 2011), the long-term speech information was utilized for unsupervised SAD under noisy conditions. Recent work in (Sholokhov et al., 2018) summarizes SAD developments in the context of semi-supervised and unsupervised techniques. This introduced the idea of semi-supervised learning in conventional expectation maximization (EM) algorithm for GMMs (Sholokhov et al., 2018).

A hard-clustering approach using sub-band log-energy features was propose by (Górriz et al., 2006) for unsupervised SAD. It performed better than standard SAD systems such as ITU-T G.729, ETSI GSM AMR and ETSI AFE on a Spanish database. Another SAD approach used long-term temporal and spectral features in a statistical model for noise robustness (Fukuda et al., 2010). It led to improved performance as compared to ETSI AFE-SAD at low SNRs (Fukuda et al., 2010). A generalized Gamma distribution based likelihood ratio test was considered for SAD in (Shin et al., 2010). It outperformed SAD based on conventional Laplacian and Gamma distributions. A variational Bayes approach for SAD employed two parallel classifiers for noise-only and speech-with-noise (Cournapeau et al., 2010). Online expectation maximization was used for simultaneous adaptation of statistical model and decision threshold (Cournapeau et al., 2010).

3.4 Baseline SAD approaches

3.4.1 Combo Feature, Combo-SAD Unsupervised SAD

The Combo-SAD feature (Sadjadi and Hansen, 2013) was developed for unsupervised SAD on the DARPA RATS corpus. The handcrafted five-dimensional features were introduced

in (Sadjadi and Hansen, 2013). The Combo-SAD refers to a five-dimensional feature system using harmonicity, clarity, prediction gain, periodicity and spectral flux (Sadjadi and Hansen, 2013). These five features are projected into a single feature using principal component analysis (PCA). The final one-dimensional feature named *Combo* is later leveraged for unsupervised SAD with a two-component GMM. The Combo features projected into a 1-D feature is modeled with a two-component GMM where SAD threshold is a convex combination of GMM means. The weights for convex combination can be varied to obtain the minimum detection cost function (DCF) for a given dataset (Sadjadi and Hansen, 2013). In this study, we used one-dimensional Combo features and Combo-SAD for comparative studies performed in this chapter.

3.4.2 SSGMM

The semi-supervised GMM (SSGMM) is a method that sits between supervised and unsupervised GMMs (Sholokhov et al., 2018). If all training data are labeled, the SSGMM would be the same as training independent class-specific GMMs for speech and non-speech (Kinnunen and Rajan, 2013). GMMs could be used in a supervised or unsupervised setup for SAD (Alam et al., 2014). SSGMM trains two GMMs using features from an utterance, where one GMM represents the speech and another non-speech. The study by (Kinnunen and Rajan, 2013) considered higher energy frames as speech and lower energy frames as non-speech where only a fraction of the highest and lowest energy frames were chosen. Thus, SSGMM required some labels that could be obtained either from human annotators or using a simpler SAD. These initial SAD labels were later used for supervised training of speech and non-speech GMMs. The core of SSGMM was generative training of separate GMMs for speech and non-speech where the initial labels were expected to be accurate for good performance under noisy conditions. This method assumes availability of reliable SAD labels for a sufficient amount of stable training data for both speech and non-speech (Sholokhov et al., 2018).

3.4.3 SohnSAD

SohnSAD was based on a robust decision rule derived from the generalized likelihood ratio test that considered the geometric mean of likelihood ratios over all frequency bins (Sohn et al., 1999). The noise statistics are obtained from the estimated noise spectrum. It used an effective hang-over scheme by considering a first-order Markov process model of the previous speech frames. The study (Sohn et al., 1999) performed experiments under low SNRs and vehicular noise where SohnSAD outperformed the G.729B speech activity detector.

3.4.4 rSAD

The rSAD was chosen as a third baseline SAD in this chapter (Tan and Lindberg, 2010). Its main features include: (i) effective frame selection based on a-posteriori signal-to-noise ratio (SNR) ; (ii) use of an efficient energy distance instead of standard cepstral distance. This method was found to be effective at low SNRs as a result of built-in speech enhancement. Initially, the audio undergoes a high-pass filtering step followed by selection of the high-energy frames based on a-posteriori SNR weighted energy-difference. Next, the pitch values are computed for each frame. The high-energy frames without a pitch value in a reasonable range are taken as non-speech. This process generated the enhanced signal by setting the high-energy invalid pitch frames to zero. The modified version of the minimum statistics method is used for estimating the noise spectrum. In the last step, the a-posteriori SNR weighted energy-difference is extracted from the enhanced signal and frames with valid pitch are detected and labeled as speech.

3.4.5 USC Neural Network SAD

This is a supervised SAD system trained on DARPA RATS data (Van Segbroeck, Tsiartas, and Narayanan, Van Segbroeck et al.). The Gammatone, Gabor, long-term spectral variability and voicing features were combined together and used for training the neural network.

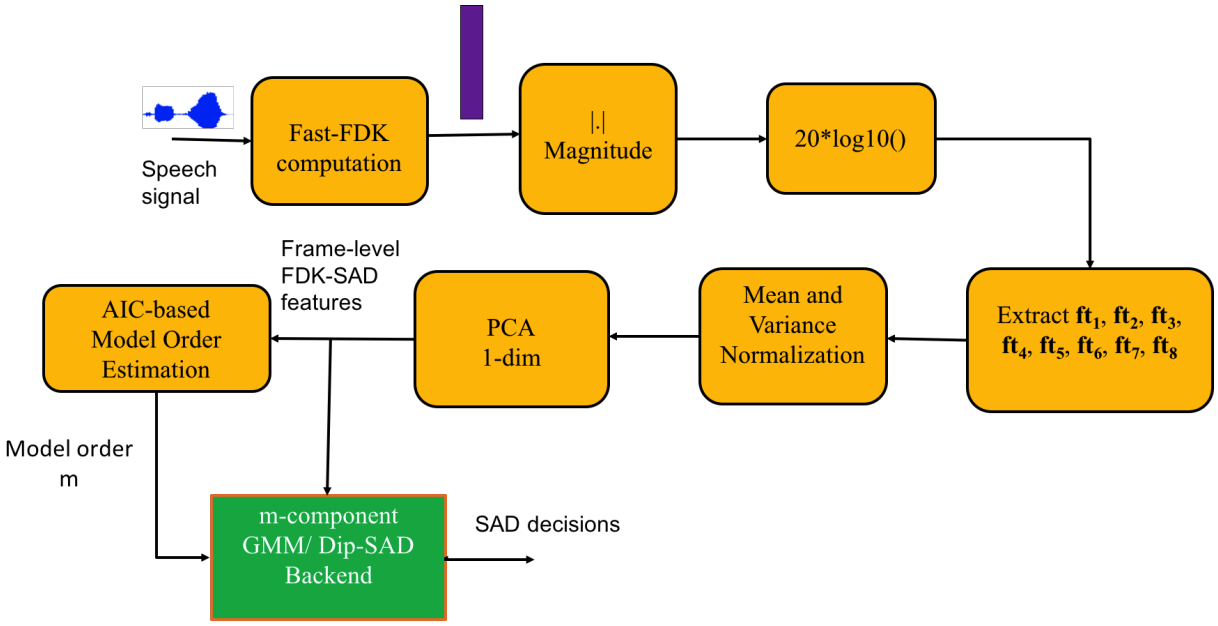


Figure 3.1. Block diagram of frequency-dependent kernel (FDK) feature extraction. These are frame-level features to be used with proposed SAD backends: (1) variable model-size Gaussian mixture model(VMGMM); (2) dip-based robust clustering (DipSAD).

The extracted features used speech characteristics such as spectral shape, spectro-temporal modulations, periodicity (pitch harmonics), and long-term spectral variability. The features used long context-windows to obtain a combined feature vector. These features were used for training a neural network (Van Segbroeck, Tsiartas, and Narayanan, Van Segbroeck et al.). The evaluation on the DARPA RATS corpus showed effective results, thus validating the applicability of developed SAD system as a useful comparison system. We adapt this system as supervised SAD baseline.

3.5 Frequency-dependent kernel features

Coming back to our discussions on multi-layer noise from Sec. 3.3.1, we knew that under adverse noisy conditions, there could potentially be many noisy processes contributing to utterance-level speech and non-speech statistics. These processes lead to the creation of noisy audio were likely to be more Gaussian than that in clean audio in accordance with the

central limit theorem (Rosenblatt, 1956). This motivated us to leverage frequency-dependent Gaussian kernels (FDGKs) for audio spectral decomposition. The choice of a Gaussian kernel was advantageous over other possible kernels due to its compact time-frequency spectrum. For Gaussian kernels, the product of uncertainty in time and frequency domains is minimum (Harris, 1969) according to the uncertainty principle. Keeping naturalistic audio streams with multi-layer noise as our target application, we settled on evaluating Gaussian kernels for SAD. There was another motivation for frequency dependence in choosing Gaussian kernels. The human hearing system is more sensitive to low frequency spectral resolutions versus higher frequency regions.

The Mel-scale was designed to loosely mimic frequency sensitivity of the human auditory system. Mel-frequency cepstral coefficients (MFCCs) based on a Mel-scale filter bank have been used in various speech systems (Davis and Mermelstein, 1980; Sahidullah and Saha, 2012). High sensitivity of the human auditory system towards low frequencies help motivate a frequency-dependent width for the analyzing Gaussian kernels.

Next, we describe the extraction of the proposed FDK-SAD features derived from post-processing of the FDK spectrum. Using Frequency-dependent Gaussian Kernels (FDGKs), we had alternate windows for filtering various frequency-bins of an input audio-frame. The Gaussian kernels have a frequency-dependent variance. We first establish the mathematical definition of the FDK spectrum denoted by $D(\tau, f, \theta)$. Let us assume that $s(t)$ is the time-domain audio signal. Next, its FDK spectrum, $D(\tau, f, \theta)$ is defined as,

$$D(\tau, f, \theta) = \int s(t) w(\tau - t, f, \theta) \exp(-j2\pi f t) dt, \tag{3.1}$$

where θ is the shape parameter of the FDK and $w(\tau - t, f, \theta)$ is the frequency-dependent kernel. For consistency, this kernel should satisfy:

$$\int w(\tau - t, f, \theta) d\tau = 1. \tag{3.2}$$

Eq. 3.2 ensures that by averaging the $D(\tau, f, \theta)$ over all time-shifts, τ produces the traditional Fourier transform spectrum. We leverage here the Gaussian kernels whose width (standard deviation) was inversely proportional to the frequency. The proposed FDGK used in this chapter is defined as:

$$w_{Gauss}(\tau - t, f, \sigma) = \frac{|f|}{\sqrt{2\pi}\sigma} \exp(-\frac{f^2(\tau - t)^2}{2\sigma^2}). \tag{3.3}$$

Using the proposed FDGK from Eq. 3.3 in Eq. 3.1, we obtain a frequency-dependent spectrum for each audio-frame. The variable τ in Eq. 3.1 represents the skip-rate vector from the windowing process (i.e., start-time for each overlapping frame). In this study, we use a 10ms skip-rate, so $\tau = [0.01, 0.02, 0.03, .., T]$ where T is the total duration of the audio signal $s(t)$ (in seconds). We estimate the FDK spectrum defined by Eq. 3.1 at discrete frequencies with a reasonable separation between successive frequency-bins. In this study, we considered analyzing the frequency vector set of $\mathbf{f_{vec}} = [40, 60, 80, .., 4000]$. Thus, we have steps of 20Hz that save computation time. We re-sampled the audio signal to 8kHz before FDK decomposition. In Eqs. 3.1, 3.2 and 3.3, the variable t refers to time (in seconds). For each 32ms analysis time-window with a 10ms skip-rate, we have two time-variables, t_i and t_f representing the start-time and end-time for the i-th window, given as:

$$\begin{aligned} t_i &= \tau_i, \\ t_f &= \tau_i + W_{size}, \end{aligned} \tag{3.4}$$

where W_{size} is the analysis window-size. For this study, the window size, W_{size} is fixed to 32ms which corresponded to 256 samples at an 8 kHz sampling rate. We define the new time-variables as follows:

$$\begin{aligned} t'_i &= t_i - \tau_i, \\ t'_f &= t_f - \tau_i. \end{aligned} \tag{3.5}$$

Using Eq. 3.4 in Eq. 3.5, we have

$$\begin{aligned} t'_i &= 0, \\ t'_f &= W_{size}. \end{aligned} \tag{3.6}$$

Thus, the new start-time and end-time variable, t'_i and t'_f respectively are constant for each time-window. After substituting $\sigma = 1$, we re-write Eq. 3.3 as:

$$\mathbf{P2} = w_{Gauss}(\mathbf{t}', \mathbf{f_{vec}}) = \frac{\mathbf{f_{vec}}}{\sqrt{2\pi}} \exp\left(-\frac{\mathbf{f_{vec}}^{.2} \times \mathbf{t}'^{.2}}{2}\right), \tag{3.7}$$

where $\mathbf{t}' = 0 : 1/F_s : 32ms$ is constant for all time-windows (audio frames) where F_s is the 8kHz sample rate. In Eq. 3.7, $\mathbf{t}'$ is the $t - \tau$ vector that was specifically defined over the interval $[t'_i, t'_f]$. Here, **P2** is used to represent the product-term-2 that will be used later for computation of the FDK spectrum. The square operation in Eq. 3.7 is applied element-wise, i.e., ($\mathbf{f_{vec}}^{.2}$ stands for element-wise square of $\mathbf{f_{vec}}$ and similarly $\mathbf{t}'^{.2}$ is element-wise square of $\mathbf{t}'$). Also, $\times$ represents matrix multiplication between vector $\mathbf{f_{vec}}^{.2}$ of say, dimension M x 1 and vector $\mathbf{t}'^{.2}$ of say, dimension 1 x N. Thus, w_{Gauss} is a matrix with size dimensions of M x N.

Using Eq. 3.7, we can pre-compute $\mathbf{w}_{Gauss}(\mathbf{t}', \mathbf{f_{vec}})$ and store in memory to save computational resources during processing. From Eq. 3.1, we can also see that the second term, $w(\tau - t, f, \theta)$ can also be pre-computed using Eq. 3.7 and does not need to be computed for each time-window. The remaining two terms in Eq. 3.1 correspond to the audio signal $s(t)$ (first term) and $\exp(-j2\pi ft)$ (third term). We represent the product-term-3 corresponding to $\exp(-j2\pi ft)$ for the i-th window as $\mathbf{P3_i}$ defined as,

$$\mathbf{P3_i} = \begin{pmatrix} \exp(-j2\pi \mathbf{f_{vec}} \tau_i) \\ \exp(-j2\pi \mathbf{f_{vec}} \tau_i) \\ \cdots \\ \cdots \\ \exp(-j2\pi \mathbf{f_{vec}} \tau_i) \end{pmatrix} \tag{3.8}$$

Thus, $\mathbf{P3_i}$ is a vector of dimension size M x W_{size} that can also be pre-computed using Eq. 3.8. For the purposes of discussion, suppose $\mathbf{P3}$ is an array containing all such matrices from each time-windows where $\mathbf{P3_i}$ is for the i-th audio frame. We window the audio signal $s(t)$ with a rectangular window of size W_{size} using τ as the skip-vector. Given this scenario, the product-term-1 corresponding to audio signal $s(t)$ for the i-th window denoted as $\mathbf{P1_i}$ is defined as,

$$\mathbf{P1_i} = \begin{pmatrix} s(\tau(i)) & s(\tau(i)+1) & \cdots & s(\tau(i)+W_{size}) \\ s(\tau(i)) & s(\tau(i)+1) & \cdots & s(\tau(i)+W_{size}) \\ \cdots & \cdots & \cdots & \cdots \\ s(\tau(i)) & s(\tau(i)+1) & \cdots & s(\tau(i)+W_{size}) \end{pmatrix} \tag{3.9}$$

where the subscript i denote the i-th window starting at time, $\tau(i)$ and ending at time $\tau(i) + W_{size}$. $\mathbf{P1_i}$ will therefore be a matrix of size M x W_{size}. If we let $\mathbf{P1}$ be an array containing all matrices, then $\mathbf{P1_i}$ will correspond to all audio-frames indexed by i.

It is clear from Eq. 3.9, that $\mathbf{P1_i}$ can be efficiently computed from the audio signal. Thus, for efficient computation of the FDK spectrum as given by Eq. 3.1, we perform pre-computation of Eqs. 3.7, 3.8 and 3.9. For each frame of the signal $s(t)$, the left hand side in Eq. 3.1 will end up being a vector as explained below. Taking an element-wise matrix product of all three terms defined by Eqs. 3.7, 3.8 and 3.9 we obtain,

$$\mathbf{C_i} = \mathbf{P1_i} \odot \mathbf{P2} \odot \mathbf{P3_i} \tag{3.10}$$

where $\mathbf{C_i}$ is a matrix of dimension M **x** W_{size} and $\odot$ denote an element-wise matrix multiplication. Next, we sum $\mathbf{C_i}$ over the second dimension to obtain a sum-vector of size M **x** 1. The resulting sum-vector of dimension M **x** 1 will be the i-th column of matrix, $\mathbf{D}$ as given by Eq. 3.1. Thus, for each audio-frame, we have a corresponding column in matrix $\mathbf{D}$. In this way, for an audio signal $s(t)$ with Nw number of frames, we obtain the FDK spectrum $\mathbf{D}$ of dimension size Nw **x** M where M is the length of the frequency-vector $\mathbf{f_{vec}}$. Next, we

post-process the FDK spectrum $\mathbf{D}$ to obtain the frame-level FDK-SAD features as explained below. We first take the absolute magnitude followed by an element-wise logarithm of the FDK spectrum to obtain the log-magnitude spectrum denoted by matrix $\mathbf{E}$ given as:

$$\mathbf{E} = 20 \log 10(|\mathbf{D}|), \tag{3.11}$$

where $|\cdot|$ is the magnitude operator. The log operator reduces the dynamic range leading to a more compact representation that is useful for subsequent processing steps. Taking the magnitude and logarithm is a common practice in many spectral techniques related to speech and other areas. We derive eight statistical descriptors from the log-magnitude spectrum $\mathbf{E}$, here in this study. There are two reasons for deriving these features: (i) first, the statistical descriptors quantify the variations in the log-magnitude spectrum; and (ii) the eight features have lower dimensions than the original log-spectral vectors for each frame. The eight statistical features derived from $\mathbf{E}$ are denoted as ft_1, ft_2,..., ft_8. The first, ft_1 is defined as,

$$ft_1^i = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \mathbf{E}(i, m), \tag{3.12}$$

where ft_1^i is the first feature for the i-th frame, and M is the dimension size of the frequency vector, $\mathbf{f_{vec}}$. The other features are defined as:

$$ft_2^i = \text{mean}(\mathbf{E}(i, 1:M)), \tag{3.13}$$

$$ft_3^i = \text{std}(\mathbf{E}(i, 1:M)), \tag{3.14}$$

$$ft_4^i = \text{geo-mean}(|(\mathbf{E}(i, 1:M))|), \tag{3.15}$$

$$ft_5^i = \text{trim-mean}(\mathbf{E}(i, 1:M)), \tag{3.16}$$

$$ft_6^i = \text{median}(\mathbf{E}(i, 1:M)), \tag{3.17}$$

$$ft_7^i = \max(\mathbf{E}(i, 1:M)), \tag{3.18}$$

$$ft_8^i = \min(\mathbf{E}(i, 1:M)), \tag{3.19}$$

where "mean", "std", "geo-mean", "trim-mean", "median", "max" and "min" represent the functions mean, standard deviation, geometric-mean, "trim-mean" which is the mean of the data excluding the top 5% and bottom 5% of the values (trim-mean was robust to outliers) , median, maximum and minimum, respectively. These features lie in different numerical ranges that necessitate both mean and variance normalization for each feature dimension. The normalized features are then combined via PCA to obtain a resulting one-dimensional feature named FDK-SAD. Again, Fig. 3.1 shows the overall block diagram of FDK-SAD feature extraction.

3.6 Variable model-size GMM backend

In this section, we formulate the variable model-size Gaussian mixture model (VMGMM) approach for unsupervised SAD. As previously mentioned in Sec. 3.4.1, the Combo-SAD features (Sadjadi and Hansen, 2013) were originally modeled with an output two-component GMM. Under adverse acoustic conditions such as multi-layer noise, Combo-SAD features may not always remain bimodal. For naturalistic audio streams with multiple noise-sources (such as tonal noise and non-stationary noise), assuming a bimodal distribution for Combo-SAD features represents a restriction that can potentially lead to poor performance. Instead of modeling Combo-SAD or FDK-SAD features as two-component GMMs, the VMGMM approach uses the Akaike information criterion (AIC) for estimating the model-order (Bozdogan, 2000; Akaike, 1981). Thus, the VMGMM approach estimates the model-order in an unsupervised manner without requiring SAD annotations. AIC for a given data and model is given as,

$$AIC = -2 \log \mathcal{L}(\hat{\theta}|y) + 2k, \tag{3.20}$$

where $\mathcal{L}$ represents the likelihood function, $\hat{\theta}$ the maximum likelihood (ML) estimate of the model parameters, k the number of estimated parameters, and y the data (SAD features). The AIC values for each model are estimated for the given data and model with an overall

minimum AIC value chosen as the *best model*. AIC (Akaike, 1981) is used for selecting a fixed model-order for each language-channel combination of the NIST-OpenSAD-2015 corpus selected by majority-voting among model-size for each utterance. For the NIST-OpenSAD-2015 corpus, a model-order in the range of two, three or four was found to be good choices. For some channels, there are multiple noise sources such as tonal noise, non-stationary noise, harmonic noise ; leading to a tri-modal or quad-modal distribution for the SAD features.

Once the model order m is estimated, we model SAD features from an utterance using an m-component GMM. This leads to an m dimensional mean-vector for the Gaussian components. Here, let μ_1, μ_2, μ_3, $\cdots$, μ_m be the means of the m Gaussian components, respectively. Next, the decision threshold for SAD is computed using a convex combination of m Gaussian means. The weights used for this convex combination are chosen within the range [0.1,0.9]. We use weights in steps of 0.1 in this range, thus the possible weights are from the set {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}. If w_1, w_2, w_3, $\cdots$, w_m are the chosen (developer-decided) weights for the convex combination of Gaussian means, the decision threshold is given as,

$$\tau_{mGMM} = w_1\mu_1 + w_2\mu_2 + \cdots + w_m\mu_m, \tag{3.21}$$

where the chosen weights satisfy the following constraint, (i.e., the weights must sum to one),

$$w_1 + w_2 + \cdots + w_m = 1. \tag{3.22}$$

The weights, w_i in Eq. 3.21 represent the relative contribution of the corresponding individual GMM mean, μ_i in the overall decision threshold. Theoretically, we could chose any value between 0 and 1 for each w_i.However, by varying the chosen weights in Eq. 3.21 we can obtain different decision thresholds and hence detection cost function (DCF) values, one for each combination of weights. The minimum among all DCF values is chosen as the NIST-OpenSAD-2015 *minDCF* value.

This resulting proposed VMGMM is a powerful modeling technique which results in superior performance as will be discussed in Sec. 3.10.1. There were two reasons that suggest both flexibility and robustness to VMGMM modeling: (i) firstly, the model-size can be chosen for each corpora on-the-fly; and (ii) the weights used for convex combination of GMM means can be independently optimized. While the model-size is estimated only using the features, optimizing the weights requires SAD transcripts for determining the optimized set of weights leading to minimum DCF.

Thus, superior performance of VMGMM will be shown to be partly due to transcript availability for weight optimization. In scenarios such as zero-resource speech processing, SAD transcripts might not be available and it is possible that a sub-optimal VMGMM solution results based on random weights. For zero-resource speech processing or other similar applications demanding unsupervised SAD with no further optimizations, we will show an alternate DipSAD solution. In the next Section, we present DipSAD which recursively segments the feature space into speech and non-speech modes using the Hartigan dip test. This is a non-parametric approach that neither makes any assumptions about the feature distribution unlike VMGMM, nor needs any parameter optimization.

3.7 Dip-based robust feature clustering

In this section, we present the alternate DipSAD solution that leverages recursions based on the Hartigan dip test (Hartigan and Hartigan, 1985) for unsupervised SAD. This is a non-parametric approach that needs no parameter optimization unlike weight optimization for VMGMMs. DipSAD is suitable for applications such as zero-resource speech processing where SAD transcripts may not be available for optimization of model weights. Also, this method makes no assumption on the feature distribution as well. It iteratively searches for a local maxima in the statistical dips of the feature distribution. DipSAD is a dip-based clustering

Algorithm 1 *computeDip*

Input: speech features sorted in ascending order i.e., $\mathbf{o}$=[o_1, o_2,...,o_N] where $o_1 \leq o_2 \leq ... \leq o_N$.
Output: primary modal interval [o_L, o_U], DIP and p-value, p.

Step 1: Initialize lower point o_L= o_1, upper point o_U= o_N and D = 0. Let F be the empirical cumulative distribution of features.
Step 2: Compute minorant G and majorant H of empirical distribution F in interval [o_L, o_U] (Hartigan, 1985). Let the points of contact with F are respectively, g_1, g_2, .., g_k (for G) and h_1, h_2, .., h_m (for H).
Step 3: Let d= max $|G(g_i)$ - $H(g_i)|$ > max $|G(h_j)$ - $H(h_j)|$ and the maximum occurs at $h_j \leq g_i \leq h_{j+1}$. Then, define o_L^0= g_i, o_U^0= h_{j+1}.
Step 4: Let d= max $|G(h_j)$ - $H(h_j)| \geq$ max $|G(g_i)$ - $H(g_i)|$ and the maximum occurs at $g_i \leq h_j \leq g_{i+1}$. Then, define o_L^0= g_i, o_U^0= h_j.
Step 5: If $d \leq D$, Stop and set DIP= $\frac{D}{2}$.
Step 6: If $d > D$, set D= max {

$$\sup_{o_L \leq o \leq o_L^0} |G(o) - F(o)|, \sup_{o_U \leq o \leq o_U^0} |H(o) - F(o)|$$

}, where sup is the supremum.
Step 7: Set $o_L = o_L^0$, $o_U = o_U^0$. Go to Step 2.

approach that leverages dip-based recursions on the feature stream from one utterance at a time in order to generate the corresponding SAD decisions.

Thus, it performs an utterance-level unsupervised SAD. Before presenting the DipSAD backend, we will revisit the Hartigan dip test in the following sub-section.

3.7.1 Hartigan dip test

This is a statistical test for hypothesizing the modality of a distribution (Hartigan and Hartigan, 1985). The dip test is based on the geometry of the corresponding feature distribution. In this context, "unimodality" is defined as follows: *a feature distribution is unimodal if its cumulative distribution is of convex type up to its modal interval and concave after that.* The dip test tries to fit a piecewise linear function that is convex, then

concave, to the cumulative distribution. The degree of unimodality is then decided on the basis of *goodness of this piecewise linear fit* (Hartigan, 1985). The DipSAD solution was motivated by recent success in applying Hartigan test for clustering extremely noisy data from other domains (Maurus and Plant, 2016). Application to speech processing, particularly unsupervised SAD, is novel contribution for this chapter. By comparing SAD feature dip statistics with that of a suitable reference unimodal distribution (i.e., null distribution) , a p-value can be set for the null hypodissertation. Using the significance level, $\alpha = 0.05$, we might reject or favor the null hypodissertation (unimodality) against the alternative hypodissertation (multi-modality). In this way, the dip test quantifies the empirical cumulative distribution's departure from the unimodality. Importantly, the dip (see Algorithm 1 *computeDip*) communicates the modal interval [o_L,o_U], the p-value and the DIP.

We now present the algorithmic aspects of the dip test. The DipSAD clustering is an utterance-level approach working on all frames of an utterance. First, we sort the speech feature vector, **feats**, in an increasing order. We still store the original feature vector in memory for preserving the temporal order (time-information) of the frames. Let the sorted features (observations) be $\mathbf{o} = o_1, o_2,...,o_N$ with $o_1 \leq o_2 \leq ... \leq o_N$ where N is the length of the feature vector (number of frames). All speech and non-speech modal intervals, (o_i, o_j) in the feature space would be pairs of values from $\mathbf{o}$. If N is the length of $\mathbf{o}$, the total number of possible modal intervals would be $\binom{N}{2} = \frac{N(N-1)}{2}$ (i.e., combinations obtained by choosing two values out of $\mathbf{o}$ vector). Next, for each modal interval (o_i, o_j), we compute greatest convex minorant, G of empirical distribution, F in (-∞, o_i) and the least concave majorant, H of empirical distribution, F in (o_j,∞). Let d_{ij} be the maximum distance between F and the curves G, H in the modal interval (o_i, o_j). With this, the DIP is given as

$$\text{DIP} = \frac{1}{2}\min\{d_{ij}\}, \tag{3.23}$$

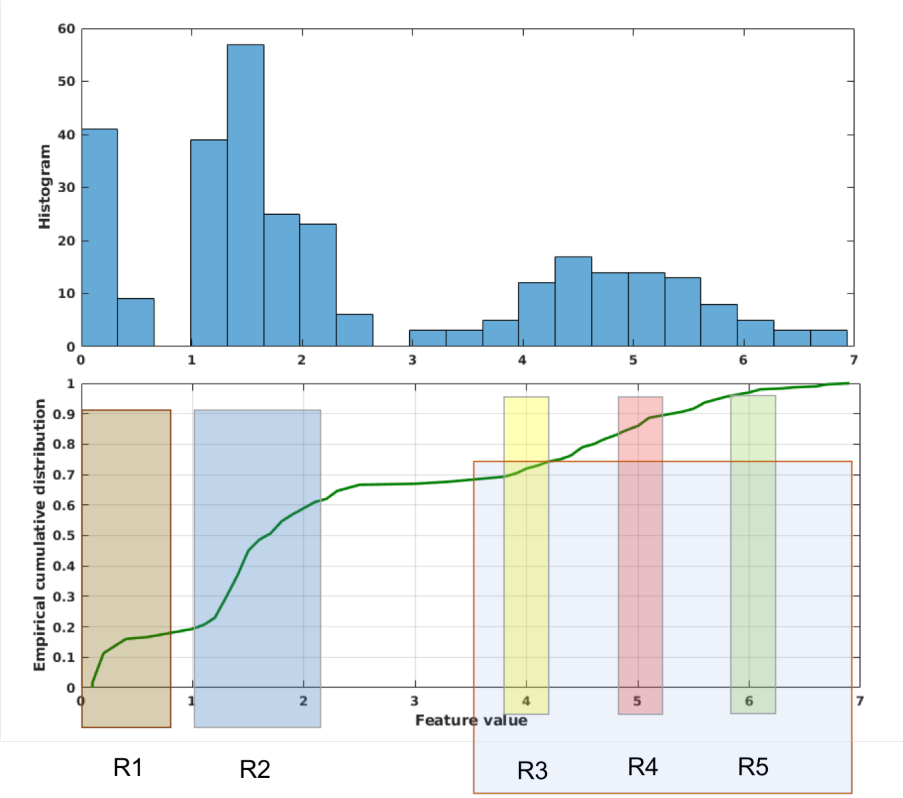


Figure 3.2. Illustration of the dip-based clustering technique on synthetic data with five classes, identified with R1 to R5 where three regions R3, R4 and R5 lie close to each other in the feature space.

over all modal interval (o_i,o_j) such that the line segment from $[o_i, F(o_i) + \frac{1}{2}d_{ij}]$ to $[o_j, F(o_j) - \frac{1}{2}d_{ij}]$ lies in the set defined by;

$$\{o, y | o_i \leq o \leq o_j, F(o) - \frac{1}{2}d_{ij} \leq y \leq F(o) + \frac{1}{2}d_{ij}\}. \tag{3.24}$$

The Eq. 3.24 ensures that the greatest convex minorant, modal segment, and the least concave majorant together form a unimodal distribution. The Algorithm 1 *computeDip* returns the DIP value, modal interval, and p-value, p from significance test for an utterance-level feature stream.

Algorithm 2 *DipSAD*

Input: speech features from an utterance
Output: speech non-speech labels

Step 1: Sort the features in ascending order and let $\mathbf{o}$=[o_1, o_2,...,o_N] be the ordered vector, where $o_1 \leq o_2 \leq ... \leq o_N$. Set significance level, α =0.05.
Step 2: $\{o_L, o_U, p\} \leftarrow computeDip(\mathbf{o})$
Step 3: If $p > \alpha$, then the detected primary modal interval is [o_L, o_U]. Else, [o_1, o_N] is primary modal interval.
Step 4: Recurse into the modal interval to find the list I_{mid} of the modal intervals within detected primary mode.
Step 5: Now, we check to the right and left of the modal interval recursively and extract additional modes if found.
Step 6: $\{u\} \leftarrow \min_{o_U \in I_{mid}} (o_U)$, $\{l\} \leftarrow \max_{o_L \in I_{mid}} (o_L)$.
Step 7: $p_l \leftarrow computeDip(\ \forall o_j : o_j \leq u)$, $p_u \leftarrow computeDip(\ \forall o_j : o_j \geq l)$.
Step 8: $I_l \leftarrow$ If $p_l \leq \alpha$, then $\forall o_j : o_j < o_L$ forms a multi-mode segment. We recurse into this interval and return all modal intervals found. Else return ϕ i.e., an empty set.
Step 9: $I_r \leftarrow$ If $p_u \leq \alpha$, then $\forall o_j : o_j > o_U$ forms a multi-mode segment. We recurse into this interval and return all modal intervals found. Else return ϕ i.e., an empty set.
Step 10: The final set of all modal interval is $I_l \bigcup I_{mid} \bigcup I_r$.
Step 11: The detected clusters were assigned to speech and non-speech classes automatically using average feature value for each cluster. As the Combo and FDK-SAD features had high positive values for speech and low values for different noises, the cluster with highest average feature value was taken as speech and rest clusters were considered non-speech. In some instances, where two prominent noise sources were present such as non-stationary background noise and occasional tonal noise, DipSAD approach created three clusters.

3.7.2 Dip-based clustering

We use the dip test recursively to locate modal intervals corresponding to speech and/or non-speech. We explain the propose clustering approach by considering Fig. 3.2 and traversing Algorithm 2 *DipSAD*. Fig. 3.2 illustrates a simulated scenario showing five categories in the feature space. Top sub-figure shows the histogram of features, while the bottom one shows the empirical cumulative distribution and depicts the dip-based segmentation. It is clear from the Fig. 3.2 that regions R3, R4 and R5 lie close to each other. Applying the approach described in Algorithm 2 *DipSAD*, the first detected modal interval consists of R3,

R4 and R5 (Step 3 in *DipSAD*). After recursing again in this interval for each o_j such that $o_L \leq o_j \leq o_U$, we obtain the three regions R3, R4 and R5 that form I_{mid} (i.e., set of middle modal intervals (Step 4)). Next, we recurse from right and left sides of the primary interval to find if other segments exist (Step 5). While recursing to the left and right, we include the nearest detected modes from the respective left or right region, (i.e., for left recursion region R3 was included in the search region while for right recursion region R5 was included (Step 6)). Thus, the upper limit (u) for the left search would be the minimum among all detected upper limits, (i.e., upper limit of region R3). On the other hand, the lower limit (l) for the right search is chosen as the maximum among all detected lower limits, (i.e., lower limit of R5 (Step 6)). This strategy ensures that the left and right searches would either be unimodal (implies same region extended in that direction such as R5 extended till the end of right region); or have multi-modalities (implies different modes in that direction such as R1 and R2 in the left). This is done in Step 6 of Algorithm 2 *DipSAD*. After we obtain the upper limit, u and lower limit, l for the left and right searches respectively, we recursively use Algorithm 1 *computeDip* on both regions to obtain the corresponding p-values, p_l and p_u (Step 7). From the p-values, we conclude unimodality if $p_l > \alpha$ and return the empty set ϕ. If $p_l \leq \alpha$, we find the corresponding modal intervals and add this to I_l such that it is the set of modal intervals in the left region (Step 8). Similarly, we perform the right search (Step 9) to obtain I_r which is the set of modal intervals in the right region. The union of the middle set I_{mid}, left set I_l and the right set I_r is the final set of all clustered modal intervals. Fig. 3.2 highlights this illustration. For SAD at the end of DipSAD clustering, we typically obtain two, three, or sometimes four clusters. If there were multiple noise sources in an utterance such as non-stationary background noise, occasional impulsive noise etc., each non-speech region will correspond to a noise-type as a separate clustered centroid. Such a scenario could be referred to as having "multi-layer noise" representing different sources appeared as separate regions in the feature space. The Combo-SAD and FDK-SAD features have large positive values

Table 3.1. Components of five propose methods for unsupervised speech activity detection.

System	**Features**	**Decision Backend**
Combo-VMGMM	Combo	Variable model-size GMM
FDK-VMGMM	FDK-SAD	Variable model-size GMM
Combo-DipSAD	Combo	DipSAD
FDK-DipSAD	FDK-SAD	DipSAD
FDK-DSAD	FDK-SAD	D-SAD
Combo-DSAD	Combo-SAD	D-SAD
Gamma-VMGMM	Gamma-SAD	Variable model-size GMM

Table 3.2. NIST-OpenSAD-2015 DCF (%) for all channels of Levantine Arabic (alv) with two-second collar around each speech region.

DCF	alv-src	alv-B	alv-D	alv-E	alv-F	alv-G	alv-H
SSGMM	1.71	5.87	4.79	3.15	2.20	3.28	3.78
rSAD	3.70	4.58	4.03	4.32	4.03	3.97	5.13
SohnSAD	3.06	4.55	4.28	2.96	2.73	3.01	3.93
Combo-VMGMM	2.01	8.31	5.25	4.66	3.23	1.11	2.82
FDK-VMGMM	0.98	5.12	4.67	4.13	2.67	0.54	2.68
Combo-DipSAD	2.68	13.21	6.40	5.83	4.19	1.34	3.63
FDK-DipSAD	2.63	9.61	9.73	5.35	4.46	2.95	3.06
Gamma-VMGMM	11.13	3.49	9.38	4.21	15.71	10.42	8.64

for speech and small positive or negative values when noise is present. We leverage this fact in automatic assignment of clusters to speech and non-speech classes. So, the cluster with highest average sample value is assigned to speech and remaining clusters are considered as non-speech. In the study by in (Graciarena et al., 2016), they also noticed that SAD features for NIST-OpenSAD-2015 data were significantly tri-modal on some channels and tri-modal GMMs helped in obtaining better performance (see section 6.4.1) in (Graciarena et al., 2016).

3.8 D-SAD: Cumulative Distribution based SAD

This approach is faster than VMGMM and DipSAD. It do not require development data unlike VGMM. It is based on a simple idea. We fit a straight line between first and last data point in cumulative distribution curve (CDC). This straight line interests the CDC at a

Table 3.3. NIST-OpenSAD-2015 DCF (%) for all channels of American English (eng) with two-second collar around each speech region.

DCF	eng-src	eng-B	eng-D	eng-E	eng-F	eng-G	eng-H
SSGMM	2.39	7.38	7.61	3.88	2.10	1.94	4.49
rSAD	2.61	3.84	3.61	3.46	2.71	2.22	5.32
SohnSAD	3.95	7.15	7.97	3.78	3.10	3.18	5.20
Combo-VMGMM	2.76	9.65	9.25	5.74	3.42	1.76	3.18
FDK-VMGMM	0.76	6.38	8.34	4.31	2.72	0.42	3.59
Combo-DipSAD	6.87	10.68	8.18	5.12	2.96	9.30	4.11
FDK-DipSAD	2.17	4.02	9.80	5.74	4.56	1.62	3.79
Gamma-VMGMM	11.85	3.13	7.68	4.15	15.69	11.62	11.19

Table 3.4. NIST-OpenSAD-2015 DCF (%) for all channels of Urdu (urd) with two-second collar around each speech region.

DCF	urd-src	urd-B	urd-D	urd-E	urd-F	urd-G	urd-H
SSGMM	1.78	5.95	4.12	2.70	1.98	1.73	3.83
rSAD	2.24	3.18	3.26	3.88	3.92	4.00	6.03
SohnSAD	4.63	7.15	6.05	4.32	3.68	3.63	6.09
Combo-VMGMM	2.20	7.61	5.62	4.67	4.06	0.72	2.92
FDK-VMGMM	1.40	7.00	5.80	4.69	3.78	0.42	3.28
Combo-DipSAD	4.22	5.85	5.51	5.30	5.26	3.67	4.78
FDK-DipSAD	2.19	6.56	7.75	4.32	5.32	1.21	4.04
Gamma-VMGMM	14.31	3.96	9.74	2.65	15.87	10.93	10.75

Table 3.5. Channel description for NIST-OpenSAD-2015 training data.

Channel	**Freq. Band**	**Modulation Type**
B	UHF	Narrow-band FM
D	HF	Single side-band AM
E	VHF	Narrow-band FM
F	UHF	Frequency-hopping spread-spectrum
G	UHF	Wide-band FM
H	HF	AM

unique point, which defines our threshold. Cumulative Distribution based SAD (D-SAD) is parameter free approach. D-SAD fits a straight line between first and last point in cumulative distribution curve (CDC). Lets say, $feats_{min}$ and $feats_{max}$ are minimum and maximum value of SAD features as extracted from CDC. We compute the slope of straight line connecting the points ($feats_{min}$, 0) (first point in CDC) and ($feats_{max}$, 1) (last point in CDC). We now

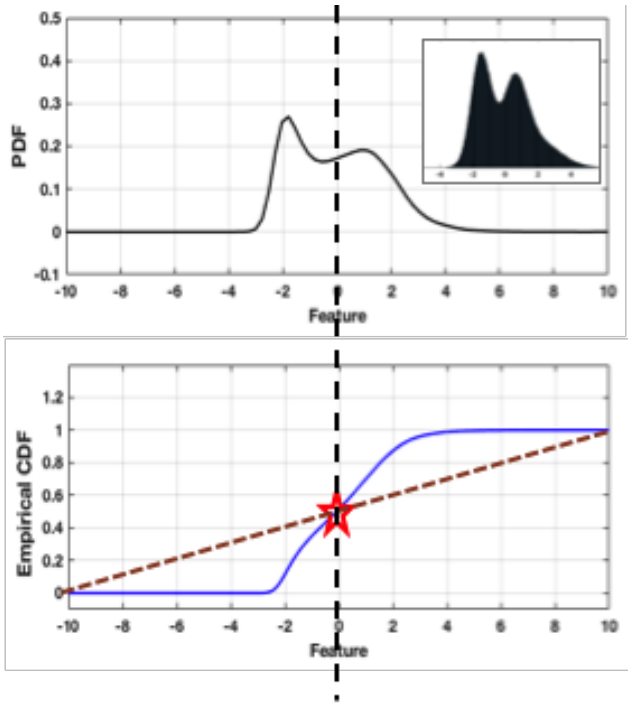


Figure 3.3. Illustrating the D-SAD decision backend. Top sub-figure shows the smoothed histogram of FDK-SAD features extracted from 80 min audio from CRSS-PLTL corpus. Bottom sub-figure shows the corresponding cumulative distribution curve (CDC). When we fit a straight line between the first point in CDC i.e., ($feats_{min}$, 0) and last point in CDC, i.e., ($feats_{max}$, 1). This line intersects the CDC at a unique point marked by red star. This point corresponds to a SAD decision threshold, $feat_{th}$ given by corresponding co-ordinate on x-axis i.e., feature axis. Thus, D-SAD is computationally simple and yet effective decision backend.

fit a straight line between these two points as given by

$$y = \frac{x - feats_{min}}{(feats_{max} - feats_{min})}, \tag{3.25}$$

where x is independent variable representing the SAD features and y is the dependent variable representing the corresponding value on CDC curve. Now, we use Eq. 3.25 for computing y for each values in range [$feats_{min}$, $feats_{max}$]. The straight line represented by Eq. 3.25 intersects the CDC curves at a point $feat_{th}$ which is the SAD threshold. At this point, $CDC(feat_{th}) = y(feat_{th})$, i.e. values of CDC and y are same. SAD features greater than $feat_{th}$ (decision threshold) are detected as speech and rest frames are classified as non-speech. Fig. 3.3 illustrates the D-SAD decision backend with FDK-SAD features extracted from 80

min PLTL session. Thus, we see that D-SAD backend is a computationally simple approach for computing the decision threshold using one-dimensional SAD features.

3.9 SAD Experiments

3.9.1 Gammatone filterbank as an alternative to FDK

The frequency-dependent kernel (FDK) could be viewed as an alternative to spectral techniques such as the Gammatone filterbank. FDK can be viewed as a class of methods such as cochlear filters (Li and Huang, 2011), Gammatone filters for large vocabulary speech recognition (Schluter et al., 2007), and auditory Infinite impulse response (IIR) filters developed for acoustic signal processing (Chi, 2003). Gammatone filters linearly approximate the human auditory processing. In this chapter, we leverage the fast estimation of Gammatone spectrum that is based on weighting the Fourier transform spectrogram. We use the Gammatone spectrogram for extracting SAD features similar to FDK-SAD extracted from the FDK decomposition (see Eq. 3.1). Here, we replace the FDK decomposition defined in Eq. 3.1 by the Gammatone spectrogram and remaining FDK pipeline remains the same. This experiment explains the benefit of FDK over Gammatone spectrogram for unsupervised SAD. It is important to note that the use of statistical descriptors extracted from the Gammatone spectrogram for SAD is novel contribution. Earlier, Gammatone filters have been used for speech recognition (Shao et al., 2009), speaker verification (Shao and Wang, 2008; Li et al., 2013) and language identification (Van Segbroeck et al., 2014). Using the Gammatone spectrogram instead of FDK in the FDK-SAD pipeline generates new features which we name *Gamma-SAD*. Prior to feature extraction, audio stream is downsampled to 8kHz. After pre-emphasizing, Hanning windowing, and framing into 32ms windows with a 10 ms skip-rate, the Fourier spectrum is obtained. We post-process the Fourier spectrogram with 64 Gammatone filters. The Gammatone spectrogram is further post-processed by a cubed-root

compression and temporal smoothing based on second-order autoregressive moving-average (ARMA filter). Finally, statistical descriptors are computed followed by mean and variance normalization before PCA processing as explained in Sec. 3.5. The final one-dimensional features extracted from the Gammatone spectrogram in the FDK pipeline are named as Gamma-SAD.

3.9.2 NIST-OpenSAD-2015 experiments

We use the NIST-OpenSAD-2015 training set for experiments reported in this chapter as techniques being evaluated were unsupervised. All seven channels from the training set are included in the experiments. Various modulation schemes such as narrow-band FM, wide-band FM, AM, frequency-hopping spread-spectrum and SSB and frequency bands such as HF, UHF and VHF were salient features of this data. We summarized the frequency bands and modulation types for OpenSAD communication channels in Table 3.5. This data was originally provided at 16 kHz sampling rate with 16 bit resolution. We downsampled it to 8 kHz for feature extraction and further processing. We evaluate the propose and baseline SAD methods on all channels of three languages namely Levantine Arabic (alv) , American English (eng) and Urdu (urd). We use 32ms analysis windows with 10ms skip-rate for extracting the SAD features that were later use in unsupervised decision backends. Table 3.1 shows the front-end features and backends for the proposed SAD systems: Combo-VMGMM, FDK-VMGMM, Combo-DipSAD, FDK-DipSAD, Gamma-VMGMM. We will discuss these results in Sec. 3.10.1. In accordance with the NIST OpenSAD protocol, we also allow for a two-second collar around each speech region. This collar region was excluded from the scoring process.

3.9.3 NIST-OpenSAT-2017 experiments

The NIST `dev` set was included with the ground-truth SAD annotations in the original corpus release, and therefore used here. According to ground-truth, this data contains an overall

41.85% speech content by duration. This data accounts for a total of 30 minutes of audio data. We considered three options: (i) no collar, (ii) a half-second collar, and (iii) a two-second collar at the beginning and end of each speech region that was accounted as collar and is not included in scoring process. This data was provided at an 8 kHz sample rate, thereby eliminating the need for re-sampling. We apply all SAD algorithms to the NIST OpenSAD corpus, so these experiments will serve as comparative studies.

3.9.4 CRSS-PLTL-II experiments

For the standalone SAD evaluations on the CRSS-PLTL-II corpus, we chose the evaluation data corresponding to a PLTL session with approximately 80-minute duration (maths course). The channel corresponding to the team lead was chosen for experiments reported in this chapter. CRSS-UTDallas human transcribers annotated the audio by hand into four classes of acoustic events described as follows. **Speech:** when just one person is speaking. **Non-speech:** when no human vocalizations are present (e.g., silences, noise, background sounds, writing-on-board sound etc.) **Overlap:** when two or more persons are speaking at the same time. **Misc:** human vocalizations that were neither speech nor overlap such as laughter, cough, lip smacks etc. Thus, the miscellaneous ("Misc") category includes everything not covered by Speech, Non-speech or Overlap events. On the basis of ground-truth annotations, we find that the total time-duration of **Speech** and **Non-speech** were the same, with both approximately being 40%. On the other hand, both **Misc** and **Overlap** events accounted for approximately 10% each of the total audio duration. The complete audio was processed individually by each SAD algorithms that include both feature extraction and decision backend. We ignore the Overlap and Misc category during the scoring phase of SAD evaluations. Thus, only speech and non-speech segments according to ground-truth were incorporated for DCF computation. Since we perform speaker diarization and behavioral speech processing in the PLTL down-stream processing, the miss-rate and false-alarms are equally important. Consequently, a 0.5

weight was given to both Pfa and $Pmiss$ for PLTL DCF computation defined as:

$$DCF_{PLTL}(\tau) = 0.5 * P_{miss}(\tau) + 0.5 * P_{fa}(\tau) \tag{3.26}$$

3.9.5 RedDots experiments: text-dependent speaker verification

We chose text-dependent speaker verification as downstream application for studying the effect of SAD on system performance. We use a subset of the RedDots corpus as described in Sec. 2.12.

3.9.6 Feature extraction

We use MFCC features extracted from 20ms overlapping windows with a 10ms skip-rate for all speaker verification experiments. We discard the DC coefficients (e.g., C_0)in MFCC output from 20 Mel filter-banks. The remaining 19 MFCC coefficients are processed with RASTA filtering in order to reduce the linear, slowly-varying channel effects (Hermansky and Morgan, 1994). Next, we append MFCC features with delta and double-delta coefficients computed over a super-window of three frames. Thus, the final feature vector size is 57 dimensional frame-level features. Finally, we use SAD for discarding the non-speech frames. This is followed by mean and variance normalization (MVN) along each feature dimension as per Eq. 3.27. Normalized features are used for training the UBM model and for MAP adaptation of the GMM-UBM speaker models.

$$\mathbf{f_{norm}} = \frac{(\mathbf{f} - \mu_{\mathbf{f}})}{\sigma_{\mathbf{f}}} \tag{3.27}$$

In Eq. 3.27, $\mathbf{f_{norm}}$ is normalized feature after MVN, $\mathbf{f}$ is original figure, $\mu_{\mathbf{f}}$ is mean vector and $\sigma_{\mathbf{f}}$ is variance. MVN helps in reducing the channel mis-match that improves the recognition accuracy.

3.9.7 Speaker modeling

Our text-dependent speaker verification system is based on state-of-the-art Gaussian mixture model (GMM) with a universal background model (GMM-UBM) (Delgado et al., 2016; Reynolds et al., 2000) for RedDots data. For this corpora, the GMM-UBM was found to perform better than methods using i-Vectors and Hidden Markov Models (HMMs) (Zeinali et al., 2016). A UBM with 512 Gaussians with diagonal covariances is trained using male data from the TIMIT train and test sets. TIMIT is a good choice for UBM training as TIMIT data is available at 16 kHz which is the sampling rate of RedDots corpus. The target speaker models, GMMs are generated by maximum-a-posteriori (MAP) adaptation of the UBM means using the enrollment data of each speaker. Let $X = \{x_n | n \in 1, 2, ..., T\}$ is set of feature vectors from enrollment data of the s-th speaker. We are provided with the trained UBM and enrollment feature vectors, X. Next, the feature vectors are aligned with respect to UBM components as:

$$\gamma_n(g) = p(g|x_n, \lambda_0) \tag{3.28}$$

where λ_0 represents the UBM (Hansen and Hasan, 2015). These $\gamma_n(g)$ are combined into a factor, $N_s(g)$ known as the zeroth order Baum-Welch statistics given as:

$$N_s(g) = \sum_{n=1}^{T} \gamma_n(g) \tag{3.29}$$

Next, the mean update of GMM-UBM speaker model is done according to following equation:

$$\hat{\mu_{\mathbf{g}}} = \alpha_g \cdot E_g[\mathbf{x_n}|\mathbf{x}] + (1 - \alpha_g)\mu_g \tag{3.30}$$

Here, α_g is defined as

$$\alpha_g = \frac{N_s(g)}{N_s(g) + r} \tag{3.31}$$

where r is known as the relevance factor. This parameter controls the influence of speaker's enrollment data on the MAP-adapted UBM-GMM speaker model. We perform only mean adaptation since it is most effective among three parameters namely weights, means and variances. We used the relevance factor of 4 for results presented in this chapter.

Table 3.6. PLTL DCF with 0.5 weight given to both Pfa and $Pmiss$ for CRSS-PLTL-II evaluation set (approx. 80 minutes). The PLTL data was corrupted at 5dB SNR with Noise n_1 and n_2 from CRSS-LDNN corpus. "Overlapped" speech and "Misc" were included in feature extraction and SAD decision making but excluded in scoring. N/A refers to not available. We skip experiments of D-SAD on PLTL with added noise as we found that all algorithms has almost similar performance on noise added to PLTL as that on original PLTL data.

System	PLTL	Noise n_1	Noise n_2
Combo-VMGMM	1.97	1.99	2.35
FDK-VMGMM	2.01	2.16	2.17
Combo-DipSAD	2.84	2.96	2.76
FDK-DipSAD	7.23	7.50	7.12
Combo-DSAD	17.68	N/A	N/A
FDK-DSAD	15.29	N/A	N/A
SohnSAD	28.20	28.51	28.71
rSAD	49.57	49.57	49.65
SSGMM	28.95	29.13	30.58

3.10 Results and Discussions

The algorithms developed in this study are aimed to provide robust unsupervised SAD for naturalistic distortions such as multi-layer noise and reverberation. In this section, we discuss the results from various experiments conducted using the proposed and baseline approaches.

3.10.1 Phase I evaluations: standalone SAD results

We five propose SAD systems as given in Table 3.1 namely Combo-VMGMM, FDK-VMGMM, Combo-DipSAD, FDK-DipSAD, Gamma-VMGMM. For standalone SAD experiments, we use OpenSAD, OpenSAT and CRSS-PLTL-II data corrupted with noise samples from CRSS-LDNN corpus.

3.10.2 NIST-OpenSAD-2015 results

We will discuss the performance of the proposed methods as compared to baseline systems namely SohnSAD, rSAD and SSGMM (as described in Sec. 3.4). NIST OpenSAD evaluation

considered missing a short speech-segment worse than introducing more false-alarms. NIST OpenSAD detection cost function (DCF) was defined as:

$$DCF_{open}(\tau) = 0.75 * P_{miss}(\tau) + 0.25 * P_{fa}(\tau) \tag{3.32}$$

where $P_{miss}(\tau)$ and $P_{fa}(\tau)$ were respectively the miss probability and probability of false-alarms. Both quantities depend on the threshold τ chosen for SAD decisions. We compute $DCF_{open}(\tau)$ for different values of the threshold τ and finally pick the minimum value referred to as $minDCF_{open}$. In this way, for each language-channel combination we obtain a $minDCF_{open}$. The DCF values are computed for each audio file and averaged to obtain the DCF for the given language-channel combination. As per the NIST OpenSAD protocol, we incorporate a two-second collar around each speech region that was excluded during scoring (Dubey et al., 2018). Tables 3.2, 3.3 and 3.4 show the *minDCF* for the proposed and baseline methods for each channel-language combination. The FDK-SAD features show superiority over Combo-SAD and other features leading to significant improvements in DCF for many language-channel combinations. In some cases, the DipSAD backend performs better than the VMGMM backend even though the VMGMM has flexibility to chose a model-order and GMM weights. This shows that DipSAD is an important technique for scenarios that lack SAD transcripts and require robust SAD. As the VMGMM backend has flexibility to choose both model-order and weights, it performs better than the DipSAD in general. The DipSAD backend does not have tunable parameters for further optimization. On the other hand, VMGMM is a model-based approach with the possibility to tune parameters for a given corpus/scenario. Table 3.2, Table 3.3 and Table 3.4 show the comparison of results obtained with Gamma-VMGMM and other baselines. The Gamma-VMGMM fails on most channels including the clean source channel unlike other methods. Gamma-VMGMM works reasonably for only channel B and E among all channels. This could be understood by reviewing Table 3.5 which list both B and E channels as having ultra or very high frequency bands with

narrow-band FM modulation. This suggests that Gamma-SAD features would work better for narrow-band FM than other modulation types. Poor performance of Gamma-VMGMM on the clean source channel and other noisy channels disprove the applicability of Gamma-VMGMM for practical SAD scenarios unless we have narrow-band FM modulation in high frequency bands. All methods other than Gamma-VMGMM achieve good performance on the clean source (src) channel and have some level of degraded performance on the other channels. SSGMM being a semi-supervised technique is competitive on many channels across the three languages. The rSAD approach is based on a in-built speech enhancement and fundamental frequency (F0) estimate, making it more effective than simply SohnSAD. In general, the DipSAD performs worse on all channels as compared to the corresponding VMGMM using the same features.

3.10.3 NIST-OpenSAT-2017 results

Table 3.7 shows the NIST DCF values (defined by Eq. 3.32) averaged for six audio recordings including PSC SSSF data *dev* with (i) no collar, (ii) a half-second collar, and (iii) two-second collar. As the quantity of actual speech data is small (i.e., 30 minute of audio) we considered different collar sizes for a comparative performance assessment of the various SAD approaches. Clearly, rSAD gives the best performance for this data, possibly due to accurate F0 estimates. SSGMM being a semi-supervised techniques performs well due to availability of reliable labels from simpler SAD. The propose FDK-VMGMM always performed better than SohnSAD due to sophisticated feature modeling using FDK-SAD and the flexible VMGMM decision backend. Once again, the VMGMM backend performed relatively better than DipSAD on both FDK-SAD and Combo-SAD features.

3.10.4 CRSS-PLTL-II results

The CRSS-PLTL-II class learning speech data is rich in non-linear distortions that are not easily quantified in terms of SNR. Particularly, it has high levels of reverberation that led

Table 3.7. DCF with no collar (DCF0) , DCF with 0.5s collar (DCF1) and DCF with 2s collar (DCF4) for NIST-OpenSAT-2017 PSC SSSF dev data.

System	DCF0 (%) [NC]	DCF1(%) [0.5 sec]	DCF4(%) [2.0 sec]
Combo-VMGMM	7.46	6.07	4.5
Combo-DipSAD	7.17	5.57	4.00
FDK-VMGMM	7.48	5.76	3.35
FDK-DipSAD	8.45	7.27	5.20
SohnSAD	8.66	6.38	3.48
rSAD	3.92	2.66	1.88
SSGMM	6.34	4.69	3.14

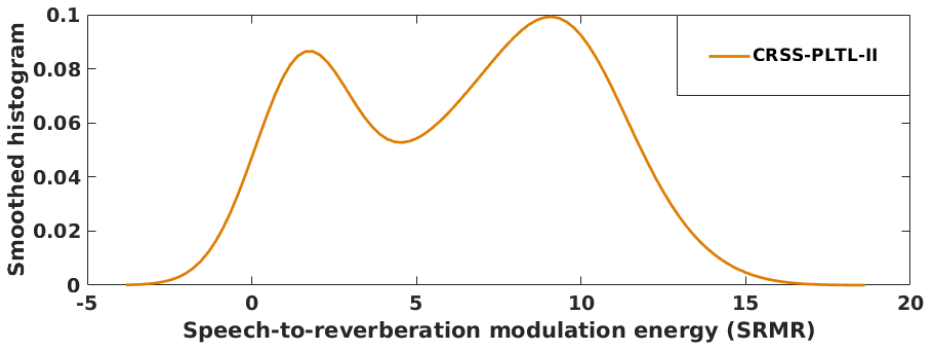


Figure 3.4. Distribution of speech-to-reverberation modulation energy ratio (SRMR) for ten-second segments of PLTL evaluation data described in Sec. 3.9.4.

to poor performance by baseline SAD methods as reported in Table 3.6. We computed the speech-to-reverberation modulation energy ratio (SRMR) (Falk et al., 2010) for ten-second segments to quantify the reverberation levels. SRMR was previously use in the REVERB Challenge (Kinoshita et al., 2016) as an objective measure of speech quality and intelligibility. It is derived from the modulation spectral components of the speech signal. Fig. 3.4 shows the smoothed histogram of the corresponding SRMR values. Here, a majority of the segments have low SRMR scores (less than 8) which show significant reverberation. Table 3.6 shows DCF values for PLTL evaluation data and PLTL data corrupted with two noise samples from CRSS-LDNN corpus. We added two noise samples from CRSS-LDNN corpus at 5 dB SNR to PLTL data. Two slices of long-duration (more than 20 minutes) noise were adopted from CRSS-LDNN noise corpus and named Noise n_1 and n_2 (see Fig. 3.5). Both samples have at least

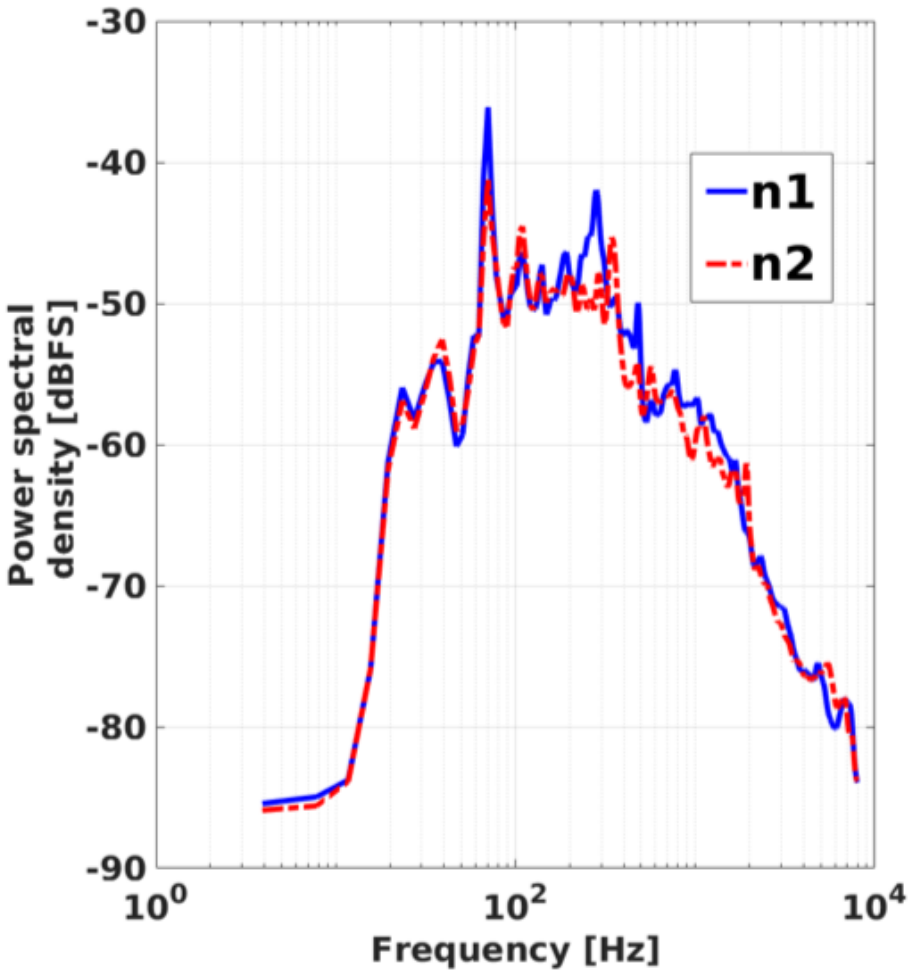


Figure 3.5. Long-term spectrum of two noise samples of duration more than 20 minutes chosen from CRSS-LDNN corpus. We can see the variety of prominent frequencies in n_1 and n_2.

two noise-sources such as construction noise, vehicle noise along with background-speakers , acting simultaneously. Clearly, the FDK-SAD features have a significant performance gain as compared to Combo-SAD features. Here, again VMGMM performs better than the DipSAD backend due to flexible modeling in VMGMM. As expected, D-SAD performance is worse than DipSAD and VMGMM. D-SAD is proposed as a computational simple decision backend that works significantly better than all state-of-the-art approaches. The performance of rSAD is dependent on fundamental frequency (F0) estimates. Due to reverberation and multi-layer noise, F0 estimates are unreliable for PLTL data leading to poor performance by rSAD. The

Table 3.8. Text-dependent speaker verification performance in terms of EER (%) on RedDots data using no SAD and different SAD approaches.

SAD Method	**EER (%)**
No SAD	7.90
SohnSAD	6.32
Combo-VMGMM	6.97
Combo-DipSAD	10.12
FDK-VMGMM	6.29
FDK-DipSAD	9.48
rSAD	6.58
SSGMM	7.37

SSGMM is based on SAD labels with models initialized on the highest and lowest energy frames. Since the highest and lowest energy frames are not a good representative of speech and non-speech in this case, SSGMM performs similar to simple baseline SohnSAD. The SohnSAD is based on signal energy and an associated hangover scheme. Both SSGMM and SohnSAD are still much better than rSAD which fails enormously due to ineffective speech enhancement and unreliable F0 estimates.

3.10.5 Phase II evaluation: text-dependent speaker verification on RedDots

The pipeline for Universal Background Model (UBM) training is based on SohnSAD and stayed the same for all experiments. Only enrollment and test trails use different SAD for comparative studies. There are a total of 35 speakers in the enrollment set. The test trails consists of 3,242 target and 120,086 non-target trails. Table 3.8 shows the effect of various SAD methods on speaker verification EER (%) for male data from RedDots part 01. The "No SAD" case refers to use of all frames in enrollment and test trails without discarding non-speech as the case with various SAD methods. FDK-VMGMM has the best performance, which is slightly better than SohnSAD. The rSAD, SohnSAD and Combo-VMGMM have comparable performances. These experiments show the superiority of FDK-SAD features and VMGMM backend for real-world data collected using mobile phones as the case with RedDots.

3.11 Summary and Conclusions

In this chapter, we introduced the CRSS long-duration naturalistic noise (CRSS-LDNN) corpus containing long-duration audio recordings of multi-layer noise. We propose frequency-dependent kernel based SAD features (FDK-SAD) and two decisions backends: (i) Variable model-size Gaussian mixture model(VMGMM) ; (ii) Dip-based robust feature clustering(DipSAD). These three advancements target different aspect of robust SAD for naturalistic audio streams. VMGMM provided a flexible decision backend. First, it modeled SAD features with a GMM whose model-size was automatically chosen based on Akaike information criterion (AIC). Secondly, SAD decision threshold was computed by a convex combination of GMM means where the weights for combining could be chosen to achieve a minimum detection cost function (minDCF). Importantly, while the choice of GMM model-order was done only using SAD features, choosing the optimum weights for minDCF assumes transcript availability. On the other hand, the DipSAD solution provides a non-parametric approach for clustering SAD features. Detected clusters were assigned to speech and non-speech classes according to the average feature value for each cluster. Unlike VMGMM, this approach did not assume any model for feature distribution and hence useful for novel acoustic scenarios with no SAD transcripts.

We studied the comparative performance of propose FDK-SAD feature and VMGMM, DipSAD decision backends with respect to state-of-the-art approaches namely SohnSAD, rSAD and SSGMM in two evaluation phases: (1) standalone SAD assessment; (2) effect of SAD on text-dependent speaker verification for RedDots data. In addition, we incorporated the Gammatone spectrogram in FDK pipeline for deriving the Gamma-SAD features. Three corpora were use for standalone SAD evaluations namely NIST-OpenSAD-2015 training set, NIST-OpenSAT-2017 public safety communications (PSC) corpus and CRSS-PLTL-II data corrupted with naturalistic noise from CRSS-LDNN corpus. We use the RedDots quarter Q4 data for studying the effect of SAD on text-dependent speaker verification. We also used the

detection cost function (DCF) metric for standalone SAD and equal error rate (EER) for text-dependent speaker verification evaluations.

During the evaluation phase I (standalone SAD assessment), SAD performance was found to be domain dependent. While FDK features had high gains for naturalistic CRSS-PLTL-II data, the improvements on NIST-OpenSAT-2017 and NIST-OpenSAD-2015 were relatively lower. The huge gap in comparative performance on CRSS-PTLT data could be understood in terms of speech features and decision models employed by various SAD methods. For the **NIST-OpenSAD** standalone SAD evaluation with two-second collar, the main takeaways were (i) VMGMM was consistently better than DipSAD for both FDK-SAD and Combo features. It was due to VMGMM's flexibility in choosing model-order and optimized decision thresholds. (ii) FDK-SAD features were consistently better than Combo features with both backends, VMGMM and DipSAD. This showed robustness of FDK-SAD features for accurate quantification of speech non-speech statistics. (iii) SSGMM being a semi-supervised approach performed worse on noisy channels than source channel. This was due to unreliable labels obtained from simpler SAD for training speech and non-speech GMMs. (iv) rSAD had the worst performance on Channel H as compared to its performance on other channels. Channel H had amplitude modulation in high frequency bands. Its performance depended to a large extent on accuracy of estimated fundamental frequency (F0). Possibly, channel H noise distorted the audio signals causing inaccurate F0 estimates. (v) SohnSAD had better performance on channel F and G as compared to the other channels. Its performance depended on signal energy and hangover scheme; (vi) Gamma-SAD features had the worst performance among all approaches. Relatively, it was better on channel B and E both having narrow-band frequency modulation.

We evaluated the **NIST-OpenSAT data** with (i) no collar, (ii) one-half second collar, and (iii) two-second collar. On NIST-OpenSAT the main takeaways were: (i) FDK-VMGMM was better than all other propose combinations i.e., FDK-DipSAD, Combo-DipSAD and

Combo-VMGMM. (ii) rSAD had best performance among all methods. Its performance depended on accurate F0 estimates that was achievable for this data. (iii) SSGMM had relatively better performance than SohnSAD as it could get reliable SAD labels for initializing the GMM models. (iv) SohnSAD had worst performance as it depended only on audio energy and hangover scheme. With unique kind of distortions in OpenSAT PSC data, the energy was not an appropriate SAD feature.

Finally, the **SAD evaluations on CRSS-PLTL-II data** had completely different takeaways. Before we disclose the takeaways, lets note salient differences between CRSS-PLTL-II data and NIST corpora. The CRSS-PLTL data was naturalistic with significant reverberation and multi-layer noise in addition to overlapped speech and "Misc" vocalizations. The speech-to-reverberation modulation energy ratio (SRMR) was less than 5 for majority of the segments showing significant reverberation. NIST-OpenSAD corpus was derived from DARPA RATS data consisting of re-transmitted telephonic conversations from source channel. Thus, the DARPA RATS data had controlled channel distortions unlike naturalistic distortions in PLTL. The OpenSAD data was not reverberant unlike PLTL. The PLTL data was collected using not-so-close wearable microphones (LENA units) creating far-field SAD scenarios. Telephonic conversations in DARPA RATS had negligible overlapped speech unlike spontaneous conversations in CRSS-PLTL leading to approximately 10% overlapped speech (according to ground-truth). This overlapped speech was present in audio stream processed by each SAD algorithm. After getting SAD labels for complete audio, we ignored the overlapped frames during DCF computation. Both false-alarms and miss-rate were equally bad for PLTL corpus given the downstream analysis consisted of speaker diarization and behavioral speech processing. PLTL DCF considered 0.5 weight for both Pfa and $Pmiss$. On the other hand, the OpenSAD assumed miss-rate to be more serious problem than false-alarms leading to 0.75 weight given to $Pmiss$, while Pfa was assigned 0.25. Main takeaways from SAD evaluations on PLTL data were as follows. (i) There were three evaluation sets derived from the PLTL

data- original PLTL, PLTL data corrupted with 5 dB noise n_1 from CRSS-LDNN corpus and PLTL data corrupted with 5 dB noise n_2 from CRSS-LDNN corpus. Performance of all SAD algorithms degraded when we moved from original PLTL data to corrupted PLTL data. (ii) Due to significant overlapped speech, state-of-the-art methods such as SohnSAD, SSGMM and rSAD broke as their performance depended on speech energy and fundamental frequency (F0) estimates. Among chosen baselines, SohnSAD and SSGMM were still doing better than rSAD on PLTL data. This was due to F0 dependence in rSAD. In the presence of reverberation, multi-layer noise and overlapped speech, it was challenging to obtain accurate F0 estimates. Furthermore, rSAD performance depended on inbuilt speech enhancement that computed noise spectrum from non-speech segments. The reverberant PLTL data cause inaccurate speech enhancement leading to worst performance. (iii) SohnSAD was dependent on speech energy and hangover scheme. Similarly, SSGMM trained speech and non-speech GMM models on highest and lowest energy frames. As such, both methods had similar performance. (iv) Proposed VMGMM backend was better than DipSAD with both Combo and FDK-SAD features. (v) FDK-SAD features were robust to both noise n_1 and n_2 leading to good performance on three datasets derived from PLTL and CRSS-LDNN corpora. (vi) DipSAD solution was non-parametric with no model assumptions unlike VMGMM. Even then, it performed much better than state-of-the-art SohnSAD, rSAD and SSGMM. It showed the effectiveness of DipSAD for zero-resource naturalistic audio streams. In this way, the results on PLTL data validated the efficacy of propose FDK-SAD features and VMGMM, DipSAD backend for robust SAD for naturalistic audio streams. D-SAD is a computational simple decision backend that works significantly better than all state-of-the-art approaches. As expected, D-SAD performance is worse than DipSAD and VMGMM as it lacks flexibility to optimize weights unlike VMGMM. Unlike DipSAD where recursions helps in refining the SAD output, D-SAD is parameter free and work in single iteration. It is worth noting that even if D-SAD is computationally simple, it significantly out-performs state-of-the-art approaches.

Now, we summarize the takeaways from **text-dependent speaker verification experiments** using propose and baseline SAD approaches. We considered "No SAD" in this pipeline. We use SohnSAD during training of UBM model that was fixed for all experiments. During enrollment and test trials for each speaker, the corresponding SAD was use. In principle, we could have use different SAD for UBM training to be more accurate however we did not choose to do that for reducing the experimental overhead. EER metric was use for comparative studies involving GMM-UBM speaker models. The main observations on these experiments were as follows. (i) VMGMM was better than DipSAD thus following the same trend as in standalone SAD evaluations. (ii) FDK-VMGMM had best performance supporting accurate modeling of short audio using FDK-VMGMM pipeline. (iii) SSGMM had worst performance given the short audio recordings having insignificant labels for training GMM models. (iv) rSAD was poorer than VMGMM and SohnSAD, however still better than "No SAD" case. Thus, rSAD had reasonable performance on RedDots as it did not need labels for training GMM models unlike SSGMM. Speech enhancement in rSAD made it a competitive SAD for RedDots.

Overall, the results on standalone SAD evaluations showed that the propose FDK-SAD features and VMGMM, DipSAD back-ends were robust for naturalistic distortions. For naturalistic CRSS-PLTL corpus, the propose approaches were significantly better than the baseline methods.

CHAPTER 4

SINCNETS BASED SPEAKER RECOGNITION AND DIARIZATION [1]

4.1 Introduction

Speaker Diarization is front-end for multi-subject speech technologies. It provides solution for *who spoke and when?* (Yella and Stolcke, 2015). It is in general an unsupervised/semi-supervised system. It consists of sub-systems: (i) speech activity detection (SAD); (ii) speaker change detection; (iii) clustering; (iv) re-segmentation where step (iv) is optional. Some approaches combined step (ii) and (iii) into joint segmentation and clustering (Anguera et al., 2012). Recently, researchers combined audio and visual cues in spectro-temporal fusion for diarization (Gebru et al., 2018). This approach suits for challenging scenarios where several speakers are engaged in interaction and assumes availability of video. Practical applications of speaker diarization (Dubey et al., 2016a) include broadcast new analysis, low-latency speaker spotting (Patino et al., 2018) and behavioral study (Dubey et al., 2017) .

State-of-the-art diarization systems use i-Vectors in speaker clustering. Recently, neural network embeddings (d-vectors) were benchmarked for diarization task. A three layer network with one LSTM layer and final linear layer was used for speaker diarization (Zhang et al., 2018). However, most deep neural network based speaker embedding extractor are trained on significantly large amount of data which is not always available (Snyder et al., 2018). Recently, CNNs were explored for deriving speech representations for a variety of tasks. Such approaches use magnitude spectrum for speech feature learning. The idea of exploring a first layer with parameterized Gaussian filters in a deep neural network was explored for speech recognition (Seki et al., 2017). It was trained at frame-level using spectrogram features (Seki et al., 2017). Some studies evaluated custom layer consisting of Gabor filters

[1] ©2019 IEEE. Portions Adapted, with permission, from H. Dubey, A. Sangwan, and J. H. L. Hansen, "Transfer Learning Using Raw Waveform SincNet for Robust Speaker Diarization," In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6296-6300, 2019.

using power-normalized spectrum as input for speech recognition (Chang and Morgan, 2014). More recently, using raw waveform for training neural network is an emerging trend. This approach is advantageous as it eliminate the feature extraction pipeline. Learning from time-domain signal showed good results for tasks such as speech recognition (Sainath et al., 2015), emotion identification (Trigeorgis et al., 2016), speaker verification (Ravanelli and Bengio, 2018) and speech syndissertation (Van Den Oord et al., 2016).

In this chapter, we investigate SincNet for speaker diarization where the first layer consists of sinc filters. Sinc-Layer learns compact band-pass filters suitable for speaker modeling. It is parameterized by cut-off frequencies of these band-pass filters. The gain of sinc filters is learned by later (convolutional and fully connected) layers in SincNet architecture (see Fig. 4.1). SincNet was developed for speaker recognition in practical scenario where small training data (few seconds/speaker) was available while the test utterance were very short (Ravanelli and Bengio, 2018). We leverage efficient SincNet in a vanilla transfer learning (VTL) setup where the SincNet was trained for frame-level speaker recognition on out-of-domain data and later trained SincNet-VTL was used for extracting speaker embeddings from in-domain data (see Fig. 4.2). We investigated several possibilities for extracting features, namely F1, F2 and F3 that were later pooled to fh segment-level speaker models. We employed length-normalized SincNet-VTL embeddings in a diarization pipeline that uses ground-truth speaker segmentation and cosine K-means clustering.

4.2 SincNet Architecture

Recently, SincNet was developed as an efficient architecture for processing raw speech waveform for speaker recognition (Ravanelli and Bengio, 2018). Fig. 4.1 shows the SincNet architecture that consists of six hidden layers, namely, Sinc-Layer, two 1D convolutional layer, and three fully connected layer. Sinc-Layer performs sinc-based convolutions on overlapping frames (200ms with 10ms skip rate) of time-domain signal. After the first Sinc-Layer, standard

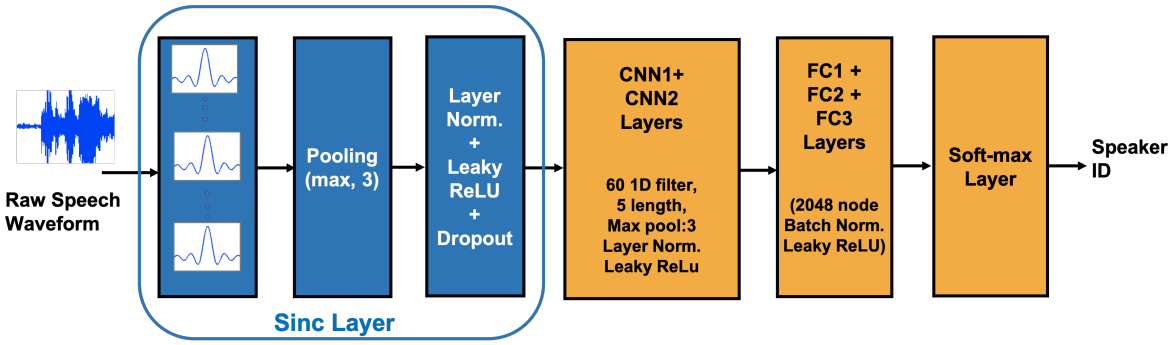


Figure 4.1. The architecture of waveform SincNet (Ravanelli and Bengio, 2018). Sinc-Layer performs time-domain convolutions on raw speech. Next, two 1D convolutional layers and three fully connected layers filter the input. Final soft-max layer perform speaker classification. Next, Convolutional and fully-connected layer are trained along with sinc layer for frame-level (200ms frames with 10ms skip-rate) speaker classification.

CNN pipeline (pooling, batch normalization, ReLU activations, dropout) were employed. As shown in Fig. 4.2, Sinc-Layer, CNN1 and CNN2 were followed by fully connected layers FC1, FC2 and FC3. Sinc-Layer has 80 sinc filters each with a length of 251 and max pooling over 3. Both CNN1 and CNN2 layers had 60 filters each with length 5 and max pooling over 3. Sinc-Layer, CNN1 and CNN2 employs layer normalization (Ba et al., 2016) and leaky ReLU activations. Three fully connected layers namely FC1, FC2 and F3 had same configuration i.e., 2048 nodes, batch normalization (Ioffe and Szegedy, 2015) and leaky ReLU activations. Final soft-max layer has number of nodes equal to speaker count in training data. This architecture takes raw speech from 200ms time-windows (frames) with 10ms skip rate and trained for speaker recognition at frame-level.

Sinc-Layer learns the formants and pitch trajectory that facilitate efficient speaker modeling (Ravanelli and Bengio, 2018) and results in compact representation. Unlike fully connected layers, convolutional ones focus on local regions of the input and extract shift-invariant features that enhances overall recognition performance. Sinc-Layer consists of parametrized sinc functions that acts as band-pass filters in spectral domain. Discrete-time sinc filters can be represented as:

$$h[m, f_1, f_2] = 2f_2 \cdot sinc(2\pi f_2 m) - 2f_1 \cdot sinc(2\pi f_1 m) \tag{4.1}$$

The $sinc(\cdot)$ functions in above equation is defined as

$$sinc(x) = sin(x)/x. \tag{4.2}$$

Thus, the Sinc-Layer tries to learn lower and upper cut-off frequencies for filters parametrized by its nodes. For results discussed in this chapter, we initialized these with the cutoff frequencies of the Mel filter-bank. Such initialization is preferred as it has more filters in lower frequency spectrum that quantifies speaker characteristics. There are two constraints in Eqn. 4.1 that need to be satisfied: $f_1 \geq 0$ and $f_2 \geq f_1$. In fact, Eqn. 4.1 is employed with the following cut-off frequencies:

$$\begin{aligned} f_1' &= |f_1| \\ f_2' &= |f_1| + |f_2 - f_1| \end{aligned} \tag{4.3}$$

From above equations, we see that Sinc-Layer tried to learn only the cut-off frequencies. Next, convolutional and fully connected layers learn the gains for each sinc filter by assigning appropriate weights. Passband ripples in Sinc filters is mitigated by Hamming windowing that smoothen the abrupt discontinuities:

$$h_w[m, f_1', f_2'] = h[m, f_1', f_2'] \cdot w_{hamming}[m], \tag{4.4}$$

where the Hamming window is defined as

$$w_{hamming}[m] = 0.54 - 0.46 \cdot cos(\frac{2\pi m}{L}). \tag{4.5}$$

The cutoff frequencies of Sinc-Layer are learned jointly with other parameters of SincNet architecture using stochastic gradient descent. SincNet is attractive for speaker modeling due to properties such as fast convergence, compact architecture (few parameters), and computational efficiency (symmetric sinc functions).

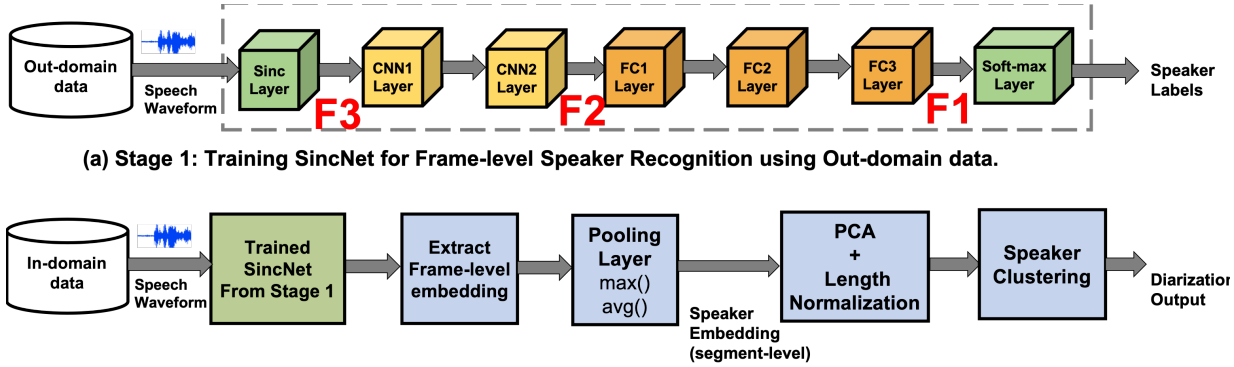


Figure 4.2. Proposed SincNet-VTL approach for extracting speaker embeddings from time-domain speech. (a) In Stage 1, SincNet is trained for frame-level speaker identification using out-of-domain data. (b) In Stage 2, we adopt the trained SincNet as feature extractor for in-domain data. We max() or avg() pooled frame-level features to obtain utterance-level embedding for getting different types of speaker embeddings. F3: output after Sinc-Layer; F2: Output after both convolutional layers; F1: Output after three fully-connected layers. These frame-level outputs were pooled to obtain the segment-level features for speaker diarization. We used these speaker embeddings for comparative study involving several speaker clustering approaches

4.3 SincNet-VTL for Speaker Modeling

This section explains the proposed approach for SincNet-based vanilla transfer learning (SincNet-VTL) as depicted in Fig. 4.2. SincNet was trained using out-of-domain TIMIT data (Garofolo, 1993). Pre-trained SincNet was adopted as feature extractor for in-domain data such as CRSS-PLTL and AMI corpora. Speaker embeddings extracted from trained neural networks are emerging alternatives to i-Vectors for speaker modeling. SincNet is a recently developed novel architecture designed for efficient processing of raw waveform (Ravanelli and Bengio, 2018). Researchers found SincNet superior to CNN for speaker recognition and verification tasks (Ravanelli and Bengio, 2018). We used SincNet trained on out-of-domain data for vanilla transfer learning (VTL). We propose to leverage out-of-domain data in speaker diarization through SincNet-VTL approach (see Fig. 4.2).

We used TIMIT corpus (Garofolo, 1993) as out-of-domain data for training SincNets. We ensured text-independent speaker modeling by not including utterance with same text for all speakers, in the training data. Non-speech at the start and end of each utterance was discarded for SincNet training. Time-domain speech signal was divided into 200ms frames with 10ms skip-rate. SincNet was trained using raw speech waveform for frame-level speaker recognition. Sinc-layer parameters was initialized with Mel-scale cutoff frequencies while rest of the network was initialized with Glorot scheme (Glorot and Bengio, 2010). Final soft-max layer implements frame-level speaker classification. The complete network was trained jointly using RMSprop optimizer with learning rate 0.001. We trained it for 360 epochs with batch size of 64. Trained SincNet-VTL has 462 nodes in output layer corresponding to speakers in training data. We tuned network hyper-parameters on TIMIT corpus. During embedding extraction on CRSS-PLTL corpus there were some segments that lasts for less than 200ms. We repeated those segments until it becomes a segment of 1s for getting speaker embedding. Since SincNet-VTL was trained on 200ms windows with 10ms skip-rate, we needed at-least 200ms for doing a forward pass on trained SincNet-VTL. We propose pooling frame-level embeddings extracted using trained SincNet-VTL for getting segment-level embeddings (see Fig. 4.2b).

4.4 Additive Margin Softmax Loss Function

In this section, we will describe the concepts related to softmax loss functions and its advanced versions. Specifically, we focus on Additive Margin Softmax (AM-Softmax) that was originally developed for face verification task (Wang et al., 2018). The main idea is to introduce a margin around decision boundary to minimize the intra-class variances. AM-Softmax is simple, intuitive, interpretable and easier to train as compared to angular Softmax (A-Softmax) (Liu et al., 2017).

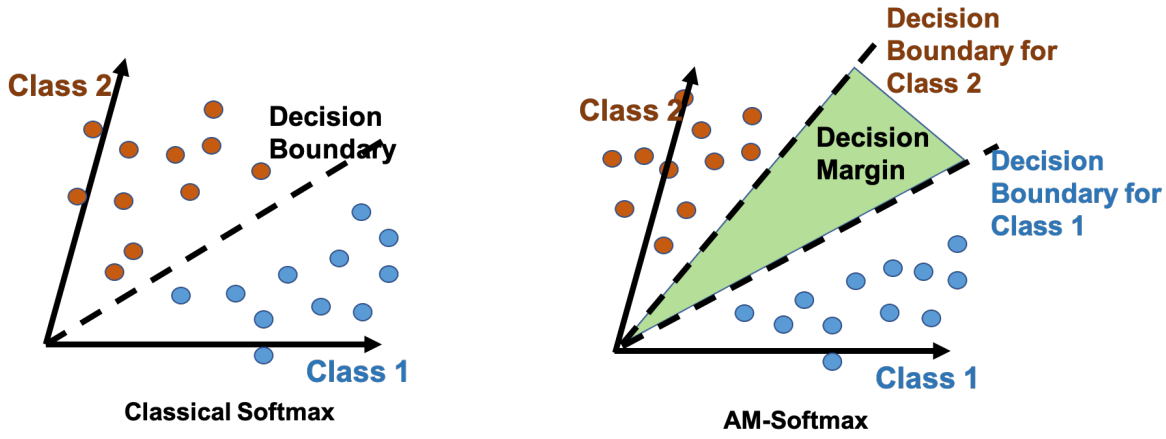


Figure 4.3. Graphical illustration of concept underlying conventional softmax and additive margin (AM-Softmax) (Wang et al., 2018).

Current state of the art deep neural networks (DNNs) for classification tasks are based on softmax loss. The softmax loss optimizes the inter-class variance. However, it lacks the capacity to reduce the intra-class variations. Several new loss functions have been studied to minimize the intra-class variance. A study proposed to penalize the feature to center distances by adding a regularization term to the loss function (Wen et al., 2016). Another way to reduce the intra-class variance is by ensuring higher gradients for well-separated data points. Researchers proposed using a scale parameter for this purpose (Liu et al., 2017; Ranjan et al., 2017; Wang et al., 2017). The scale parameter control the *temperature* of the softmax loss (Hinton et al., 2015). Another study proposed angular margin for shifting the decision boundary towards the weight vector of the corresponding class (Liu et al., 2017). Most of these studies were conducted for face verification task. However, these methods are applicable for any deep neural network (DNN) based classifier.

The softmax loss function is employed at the output layer in DNNs. Since the last fully connected (FC) layer acts as a linear classifier, the deep features of different speakers can be distinguished by their decision boundaries. Softmax creates a linear decision boundary for separating the feature vectors belonging to different speakers. It optimizes the decision boundary but do not minimize the intra-class separation/distance. Conventional softmax is

defined as:

$$L_{softmax} = -\frac{1}{n}\sum_{i=1}^{n}\log\left\{\frac{e^{W_{y_i}^T f_i}}{\sum_{j=1}^{C} e^{W_j^T f_i}}\right\} \tag{4.6}$$

Eq. 4.6 can be written as:

$$L_{softmax} = -\frac{1}{n}\sum_{i=1}^{n}\log\left\{\frac{e^{||W_{y_i}||\cdot||f_i||\cdot\cos(\theta_{y_i})}}{\sum_{j=1}^{C} e^{||W_j||\cdot||f_i||\cdot\cos(\theta_j)}}\right\} \tag{4.7}$$

Here, f is the input of last FC layer, W_j is the j-th column of last FC layer. $W_{y_i}^T f_i$ is target logit for i-th data point. The mini-batch size is n and number of speakers is C.

AM-Softmax is more interpretable and advanced version of softmax and introduces angular margin in decision boundary. It has outperformed other competitive state-of-the-art approaches such as A-softmax for face verification task (Wang et al., 2018). It introduces an additive margin such that the feature vectors belonging to same class come closer to each other and those belonging to different classes get farther from each other. Thus, it maximizes inter-class separation and minimized intra-class separation (Wang et al., 2018). AM-Softmax propose a specific $\psi(\theta)$ for introducing the additive margin, given as

$$\psi(\theta) = \cos(\theta) - m, \tag{4.8}$$

where, m is the margin parameter and usually an integer greater than 1. AM-Softmax is implemented by normalizing the features and weights. Thus, now the input becomes $x = \cos(\theta_i) = \frac{W_{y_i]}^T f_i}{||W_{y_i}||\cdot||f_i||}$. In this way, the forward pass require following computation:

$$\Psi(x) = x - m. \tag{4.9}$$

Eq. 4.9 shows that we don't need to compute gradients in back-propagation as $\Psi'(x) = 1$. This significantly simplifies the computations. The use of cosine similarity for comparing two speaker embeddings implies us to do both feature and weight normalization in the inner product layer to build a cosine layer. For AM-Softmax, the norm of both W_i and f are normalized to 1, i.e., $||W_{y_i}|| = 1$ and $||f_i|| = 1$. Next, the cosine values are scaled using a

hyper-parameter s (Liu et al., 2017; Wang et al., 2017). Now, the AM-Softmax loss function becomes

$$L_{ams} = -\frac{1}{n}\sum_{i=1}^{n}\log\left\{\frac{e^{s\cdot(\cos(\theta_{y_i})-m)}}{e^{s\cdot(\cos(\theta_{y_i})-m)}+\sum_{j=1,j\neq y_i}^{C}e^{s\cdot\cos(\theta_j)}}\right\}, \tag{4.10}$$

Eq. 4.10 can be written as

$$L_{ams} = -\frac{1}{n}\sum_{i=1}^{n}\log\left\{\frac{e^{s\cdot(W_{y_i}^{T}\cdot f_i-m)}}{e^{s\cdot(W_{y_i}^{T}\cdot f_i-m)}+\sum_{j=1,j\neq y_i}^{C}e^{s\cdot W_{y_i}^{T}\cdot f_i}}\right\}, \tag{4.11}$$

Even if scaling parameter, s can be learned through back-propagation, fixing the value of s increases the training speed. Hence, we fix s =30 for all results discussed in this chapter. Earlier, researchers found that AM-Softmax makes face recognition network to converge quickly for any reasonable value of scaling parameter, s and margin parameter, m (Wang et al., 2018). Thus, our AM-Softmax applies feature normalization and uses global scaling factor, s.

Large margin parameter, m can further reduce the intra-class variance in AM-Softmax. Also, it easily converges as compared to A-Softmax when suitable scaling parameter, s is set (Liu et al., 2017). Thus, AM-Softmax do not require huge efforts on hyper-parameter tuning and still bring large margin robustness to loss function. Fig. 4.3 shows geometric illustration of conceptual difference between conventional softmax and AM-Softmax. It shows 2-D features that lie on a circle after normalization. In this case, the decision boundary of the conventional softmax is given by a point connecting to center of the circle (origin) to a curve joining two end of the feature values. On the other hand, AM-Softmax allows for a margin between two different decision boundaries for both classes. In fact, the decision boundary becomes a marginal region for AM-Softmax.

4.5 Center Loss (CL)

In this section, we aim to enhance the discriminative power of the SincNet learned speaker embeddings by using a supervised center loss (CL). It was originally developed for face

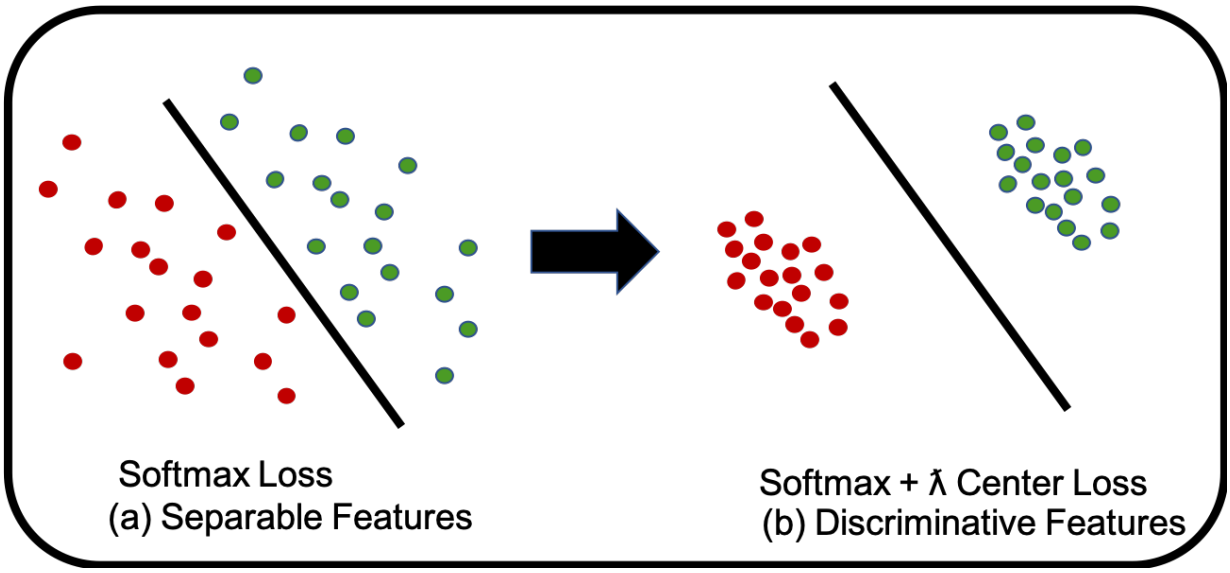


Figure 4.4. Showing that center loss (CL) enhances the discriminating power of deep speaker embedding vectors.

recognition task where it performed best when combined with conventional softmax loss function (Wen et al., 2016). It simultaneously learns a center for deep embeddings of each speaker and penalizes the distances between deep features and their corresponding class centers. It is easily trainable loss function. We leveraged CL jointly with conventional softmax and AM-Softmax for training robust SincNet based speaker classifiers. There are two benefits in this approach: (i) inter-class dispension, and (ii) intra-class compactness. Both these factors enhances speaker recognition and diarization performance.

CL aims to improve the discriminative power of DNN based speaker embeddings by minimizing the intra-class variance and ensuring the separability of different classes. The CL loss function is given as

$$L_{center} = \frac{1}{2}\sum_{i=1}^{n} ||x_i - c_{y_i}||_2^2, \tag{4.12}$$

where $c_{y_i} \in \mathbb{R}^d$ denotes the centroid (center) of y_i class and x_i is deep embedding. Eq. 4.12 effectively characterizes the intra-class variations. Ideally, the c_{y_i} need to be updated as the deep features changes. For this to happen, we require the entire training set and average the features of every class in each iteration. This is inefficient and impractical rendering CL to

be not used directly. So, the CL is used jointly with softmax as given below:

$$L_{total} = L_{softmax} + \lambda L_{center}, \tag{4.13}$$

where λ is a real number parameter used for balancing these two loss functions. If we set $\lambda = 0$, joint supervision loss function L_{total} becomes equal to conventional softmax loss. Similarly, we combined center loss with AM-Softmax for SincNet to obtain AM-CL-Softmax given by:

$$L_{total} = L_{ams} + \lambda L_{center}. \tag{4.14}$$

If we choose a suitable value of λ, the discriminative speaker embeddings from SincNet can be significantly enhanced. Earlier, researchers found that deep features from CNN based on joint supervision of CL and conventional softmax were discriminative over a wide range of λ (Wen et al., 2016). In this study, we performed experiments with different values of λ to optimize the network for diarization task.

In case of CL, joint loss L_{total} is necessary to achieve discriminative features. Lets say we use only conventional softmax loss, then the speaker embeddings will have large intra-class variance. In case, we use only CL, then the speaker embeddings and cluster centers both will be very close to zero. Thus, singly either of them is not useful but joint loss achieves good performance on classification tasks.

4.6 AM-SincNet, CL-SincNet and AM-CL-SincNet

Earlier in Sec. 4.2, we leveraged standard SincNet trained using Softmax loss function. The choice of loss function is crucial for ensuring enhanced performance of deep learning systems. Softmax is most commonly used loss function for classification tasks. However, it is not the best choice for training SincNets for speaker identification. When we replace the Softmax in SincNet by AM-Softmax loss function, we get the AM-SincNet architecture. Previously, AM-SincNet was used for speaker verification experiments, in this chapter we

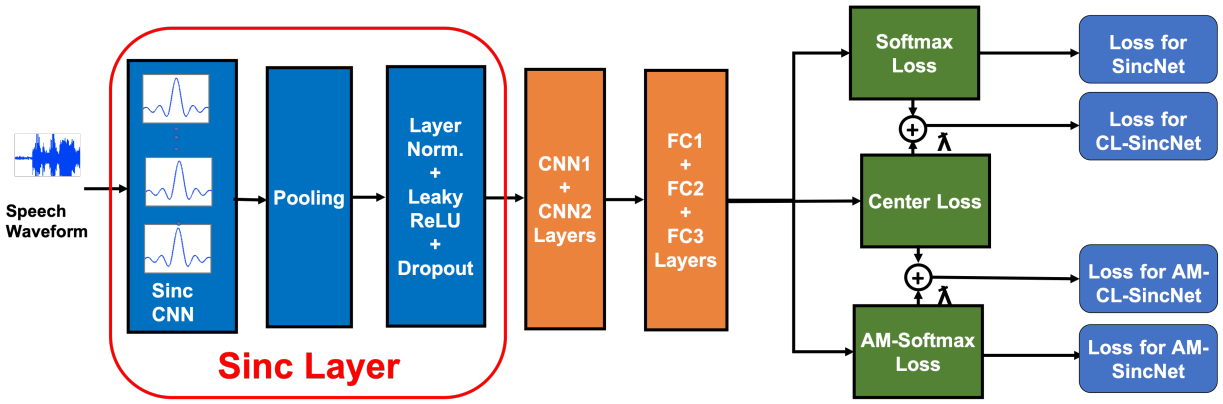


Figure 4.5. We combined conventional Softmax and AM-Softmax with Center Loss (CL) to obtain four version of SincNet: (i) standard SincNets with conventional Softmax loss; (ii) AM-SincNet with AM-Softmax loss; (iii) CL-SincNet with weighted sum of conventional Softmax loss and CL; (iv)AM-CL-SincNet with weighted sum of AM-Softmax loss and CL. All the four architecture are trained on TIMIT data (462 speakers) for frame-level speaker classification.

employ AM-SincNet in a speaker diarization task while training on out-of-domain data. The additive margin in AM-SincNet decision boundary suits well to speaker identification task as compared to conventional Softmax.Researchers compared AM-Softmax with Softmax loss function for training SincNet on speaker identification task and found that AM-Softmax was significantly better (Nunes et al., 2019) than conventional Softmax. AM-SincNet lead to relative improvement of 40% in the Frame Error Rate (FER) when trained and tested on TIMIT corpus. In this study, we leverage AM-SincNet for diarization task. We use both Librispeech (Panayotov et al., 2015) and TIMIT (Garofolo, 1993) datasets for performing experiments.

We use AM-Softmax loss function in SincNet for reducing the distance between embeddings from same speaker which enhances the speaker recognition accuracy. Fig. 4.3 graphically illustrates the idea of AM-Softmax as compared to conventional softmax. Clearly, AM-Softmax makes the speaker embeddings more discriminative as compared to conventional softmax. When CL is combined with conventional softmax in SincNet architecture, we get CL-SincNet. When CL is combined with AM-Softmax, we get AM-CL-Softmax. Thus, we have four sinc

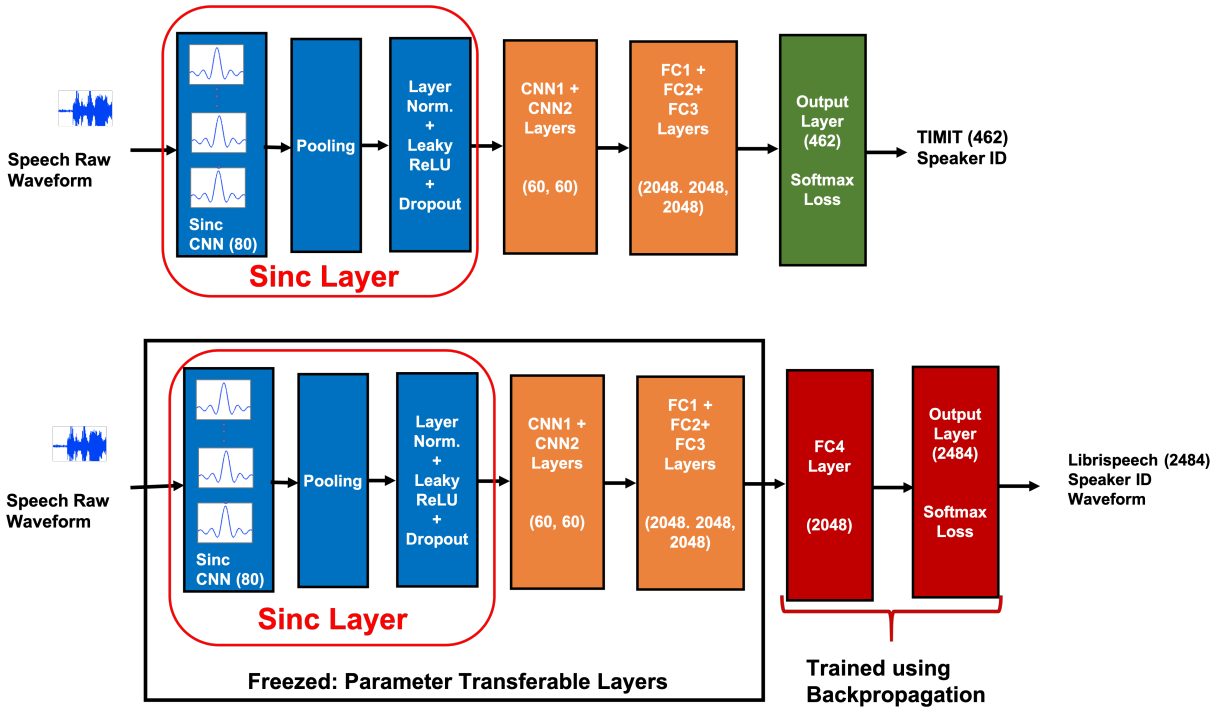


Figure 4.6. Showing that center loss (CL) enhances the discriminating power of deep speaker embedding vectors.

convolutional architecture namely: (i) SincNet, (ii) AM-SincNet, (iii) CL-SincNet, and (iv) AM-CL-SincNet. We train these four architecture for frame-level speaker recognition using TIMIT and Librispeech corpus. The trained networks for used in vanilla transfer learning (VTL) setup shown in Fig. 4.2 for extracting frame-level features that are later pooled to obtain segmental-level speaker embeddings. SincNet embeddings are combined with three proposed clustering approaches for robust speaker diarization.

4.7 Supervised Transfer Learning (STL) with SincNets

Supervised Transfer Learning (STL) refers to use of additional source of information (from out-of-domain source task) into an supervised classification on target domain (Pan and Yang, 2009). Our goal in using STL setup for SincNet is to improve learning in the target diarization task by leveraging knowledge from the source out-of-domain speaker recognition

task. STL-SincNet proposed in this sections, transfers speaker discriminative information from TIMIT to Librispeech corpora. Some of the benefits offered by STL approach are: (i) better initial performance on Librispeech speaker recognition; (ii) faster speed of convergence; (iii) better final performance on Librispeech speaker recognition; and (iv) better unsupervised transfer to Diarization pipeline for CRSS-PLTL and AMI data.

Earlier, in Sec. 4.3, we proposed trained SincNet for extracting speaker embeddings from in-domain CRSS-PLTL or AMI data. Here, the SincNet was trained only on TIMIT data. In this section, we propose supervised transfer learning (STL) for proposed novel SincNet architectures namely standard SincNet, AM-SincNet, CL-SincNet and AM-CL-SincNet. STL approach is illustrated by Fig. 4.6. Top sub-figure shows a standard SincNet that is trained for frame-level speaker classification using TIMIT data. After this network is trained, we adopt it for next stage shown in bottom sub-figure. In second stage as shown in bottom sub-figure, we discard the output layer learned in first stage. Next, we add two new layer where the layer one is output later with 2484 nodes corresponding to 2484 speakers in Librispeech data. Now, the new network is trained on Librispeech data for frame-level speaker classification. In this setup, we freeze the inner layers of pre-trained network and only the last two layers are learned using Librispeech data. This STL strategy increased the convergence speech of our network and achieved state-of-the-art results on speaker recognition for Librispeech data. There are two reasons behind this STL architecture. Firstly, the early layers in SincNet learn robust, domain-invariant features. Secondly, later layers learn speaker discrimination for speakers contained in training data. The STL-SincNet trained in this manner can be used for extracting speaker embeddings from either last layer or second last layer. These speaker embeddings can be used in diarization pipeline in same as the SincNet-VTL embeddings. In this section, we first train the model on TIMIT with ground-truth SAD and later it was trained on Librispeech without SAD. Thus, we pre-train on TIMIT data and fine-tune on Librispeech data. We only utilized standard SincNet in this task. However, other architectures such as AM-SincNet, CL-SincNet and AM-CL-SincNet can be similarly used in this setup.

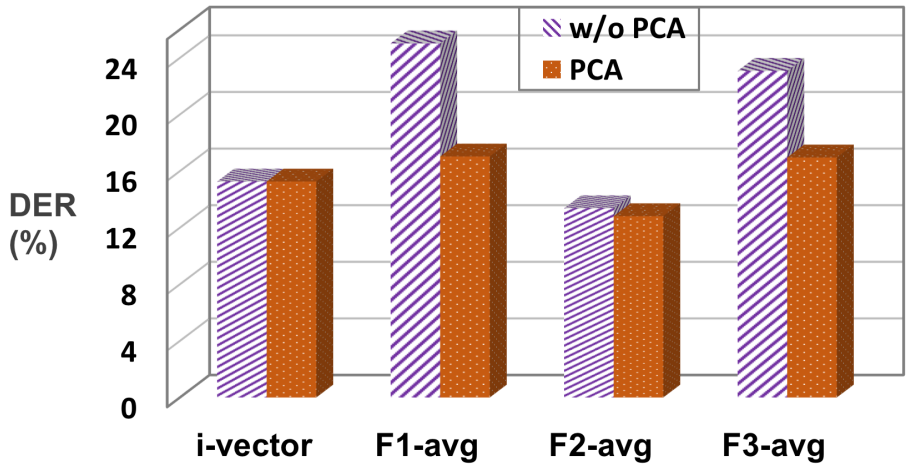


Figure 4.7. Standard SincNet trained on TIMIT data with no SAD. Evaluation results shown for PLTL data. We compare i-Vector with average pooled F1, F2 and F3 embeddings. w/o PCA means without PCA based dimensionality reduction.

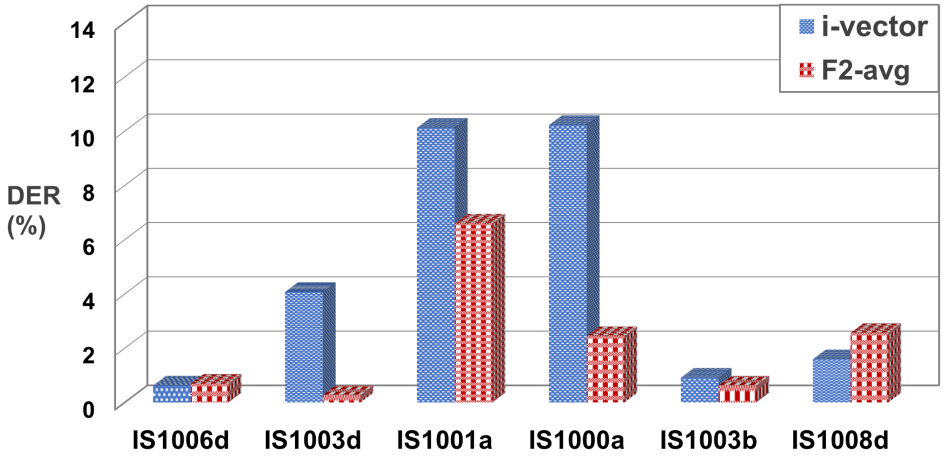


Figure 4.8. Standard SincNet trained on TIMIT data with no SAD. Evaluation results on AMI data (6 meetings). F2-avg with PCA (51 dim.) shows significant improvements over i-Vector with PCA (51 dim).

4.8 Experiments, Results and Discussions

4.8.1 Experimental Setup

Unlike NIST RT evaluations (NIST NIST, d), no forgiveness collar was allowed during scoring for results presented in this chapter. We adopted the NIST md-eval scoring script (version-22)

for DER computations. We kept all audio data at 16 kHz for experiments reported in this chapter. We trained an i-Vector extractor on TIMIT using ground-truth SAD. Since main focus of this chapter was to develop a speaker model for diarization, we used ground-truth speaker segmentation information. In this chapter, we adopted 75-dimensional (dim.) i-Vector as many segments in PLTL were approximately 1s duration (or shorter). SincNet speaker embeddings has dimensions: F1 (462), F2 (2048), F3 (6420). For all our experiments reported here, we perform length-normalization of i-Vectors/embeddings followed by cosine K-means clustering with cosine similarity. For some experiments, we employed principal component analysis (PCA) for dimensionality reduction to 51.

Table 4.1. Standard SincNet trained on TIMIT data with no SAD. We show diarization performance on CRSS-PLTL data: Effect of PCA (51 components) on DER (%) for i-Vector, F2-avg and F2-max features.

	i-Vector	**F2-avg**	**F2-max**
w/o PCA	15.26	13.37	43.55
PCA	15.26	**12.81**	14.36

4.8.2 Results and Discussions

Table 4.1 shows the effect of PCA (51 dimension) on DER (%) for three features: i-Vectors, F2-avg and F2-max where the latter two are average and max pooled version of frame-level F2 embeddings from trained SincNet-VTL network (see Fig. 4.2). Fig. 4.7 shows the comparison of i-Vector with average pooled F1, F2 and F3 embeddings with PCA. Fig. 4.8 shows DER for 6 meetings of AMI meeting corpus using i-Vector baseline and best proposed feature, i.e., F2-avg. Looking at Table 4.1, we see that i-Vector did not get DER improvements from PCA as i-Vectors were already lower dimensional. We see F2-max embeddings has benefited the most with PCA. Even if F2-avg has obtain relatively small reduction in DER with PCA as compared to F2-max, we get the best DER using PCA on F2-avg embedding. After this point, we stick to average pooling as it was better than max pooling for all the three

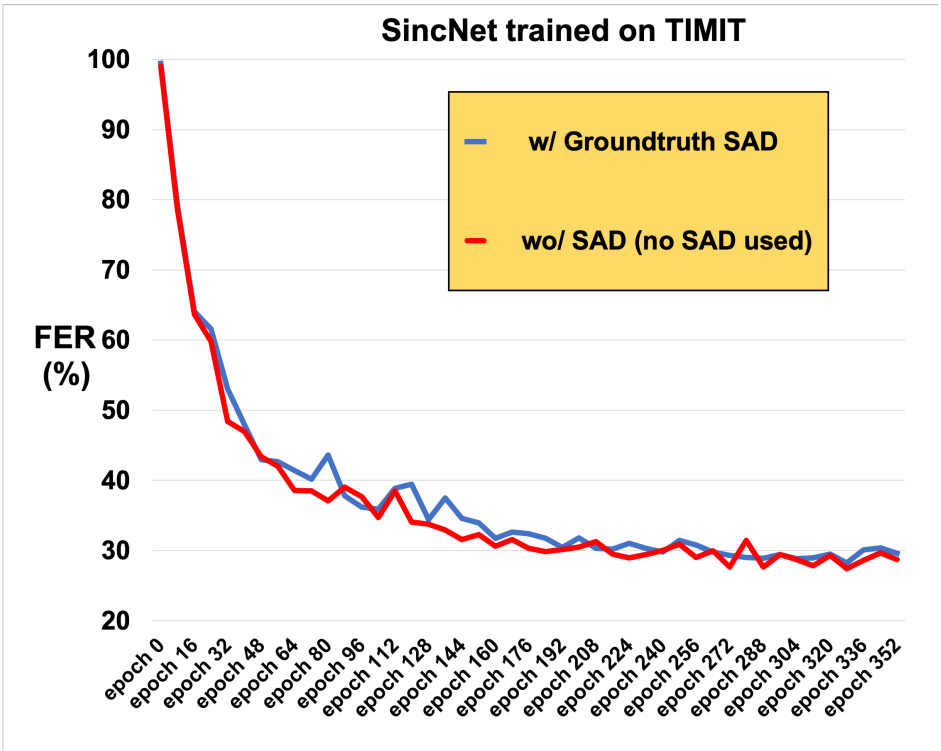


Figure 4.9. Comparing the impact of SAD on standard SincNet training in two setups: (i) Ground-truth SAD used train SincNet; (ii) No SAD used in training SincNet.

embeddings. Fig. 4.7 shows that F2-avg is best feature for speaker diarization leading to absolute and relative improvements of 2.45% and 19.12%, respectively with respect to i-Vector baseline. These comparison were done on PLTL evaluation set as it is out target domain. Fig. 4.8 shows comparison of F2-avg with i-Vector features for AMI data. Proposed F2-avg embeddings gave significant DER (%) improvement as compared to i-Vector baseline on AMI data. On average, F2-avg leads to absolute and relative improvements of 2.39 % and 52.06%, respectively over i-Vectors. In this chapter, two type of pooling operations were performed on frame-level embeddings to obtain segment-level features: max() and avg(). While max() pooling pick maximum value along each feature dimensions, avg() pooling averages along each dimension. As the results show in last section, avg() pooling performs better than max() for all three types of speaker embeddings.

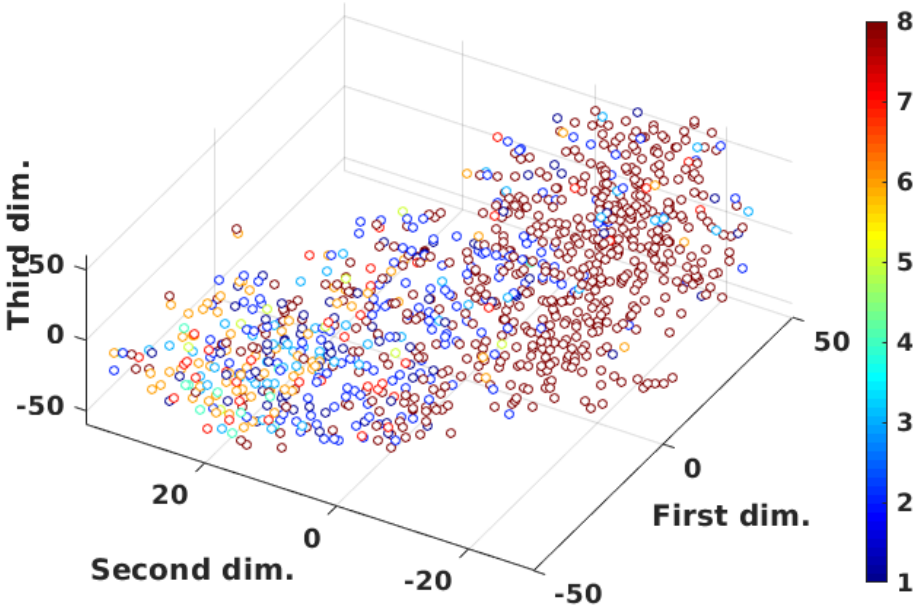


Figure 4.10. Standard SincNet trained on TIMIT data with no SAD. We extract F2-avg embeddings using ground-truth segmentation of CRSS-PLTL data. It shows t-SNE plot of F2-avg embeddings. All eight speakers are distinct while speaker ID 8 (peer-leader) spoke in most segments. The color bar looks continuous but in fact the speaker IDs are integers from 1 to 8.

Fig. 4.10 shows the t-SNE visualization (Maaten and Hinton, 2008) of F2-avg speaker embeddings for CRSS-PLTL data. The t-SNE tries to find faithful representation of high-dimensional data into a low dimensional space (typically 2D or 3D). It is a non-linear mapping that adapts to the underlying data. We employed principal component analysis (PCA) for dimensionality reduction of speaker embeddings and i-Vectors. We choose PCA with 51 components for both CRSS-PLTL and AMI evaluation sets. Since comparative studies in this chapter were focused on speaker modeling the diarization pipeline consists of ground-truth speaker segmentation and uses cosine K-means clustering with cosine similarity. SincNet-VTL embeddings (F1/F2/F3) or i-Vectors were extracted from all segments for speaker modeling. We always perform length normalization of speaker feature just before clustering. Some experiments had PCA-based dimensionality reduction before length-normalization.

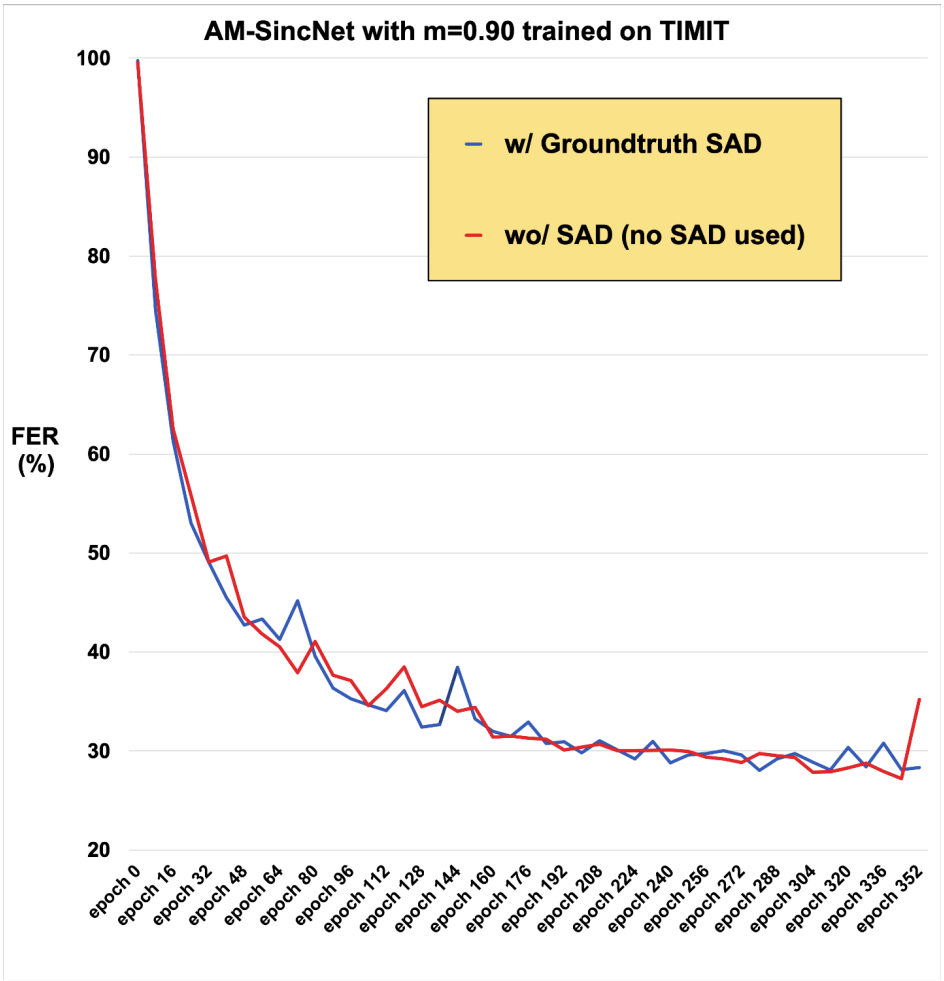


Figure 4.11. Comparing the impact of SAD on AM-SincNet training in two setups: (i) Ground-truth SAD used to train AM-SincNet; (ii) No SAD used in training AM-SincNet.

Fig. 4.9 shows that speaker recognition Frame error rate (FER) remains similar towards end of training/convergence in both cases. Thus, impact of SAD on in-domain speaker recognition is not significant. However, the corresponding DER (%) on CRSS-PLTL data was 15.42% when using standard SincNet trained on TIMIT data without SAD. On the other hand, when we used ground-truth SAD for training standard SincNet on TIMIT data, corresponding PLTL DER becomes 10.63%. Thus, using ground-truth SAD during SincNet training helped in reducing DER on out-of-domain PLTL DER. In this way, for handling with domain mis-match and/or transfer learning, use of SAD for training initial model is useful.

Fig. 4.11 shows the impact of SAD on AM-SincNet with margin parameter (m) = 0.90. Similar to above experiments, we trained AM-SincNet with and without ground-truth data using TIMIT data. Once again, we see that speaker recognition Frame error rate (FER) remains similar towards the end of training/convergence. However, the corresponding PLTL DER are 8.04 % (without SAD) and 12.10 % (ground-truth SAD). Here, we see that margin helps in making the model robust. On the other hand, we observe that not using SAD for AM-SincNet with 0.9 margin gives better results on PLTL DER than with ground-truth SAD. High margin helps in making the well separated decision boundaries that can handle presence of undesirable non-speech frames.

Fig. 4.12 shows the PLTL DER when different margin parameters, m were used for training AM-SincNet on TIMIT data without SAD. We found that among the chosen margins, 0.9 gave best results in terms of PLTL DER. However, we are not able to see any consistent trend in DER for increasing or decreasing margins. All these experiments use cosine K-means clustering.

Fig. 4.13 illustrate the impact of margin parameter in AM-CL-SincNet training for PLTL speaker diarization. We performed some selected experiments to explore the PLTL DER for different parameter set. We can see that the margin helps in making the model robust while still preserving the discriminative power of center loss (CL). PLTL DER for CL-SincNet trained with different CL parameter (λ) value is shown where 0.10 gives the least DER. We also trained the AM-CL-SincNet with fixed margin $m = 0.5$ and different CL parameter (λ) values of 0.4, 0.5 and 0.6. We also show the result from standard SincNet (right most bar) for comparison purposes. Clearly, CL-SincNet and AM-CL-SincNet are superior to standard SincNet in terms of PLTL DER. All these experiments used TIMIT data for training SincNet architecture and cosine K-means for speaker clustering in PLTL diarization pipeline. These experiments also show that the CL parameter (λ) and margin parameter (m) need tuning on target domain for best DER. We observe no consistent trend in DER with increasing

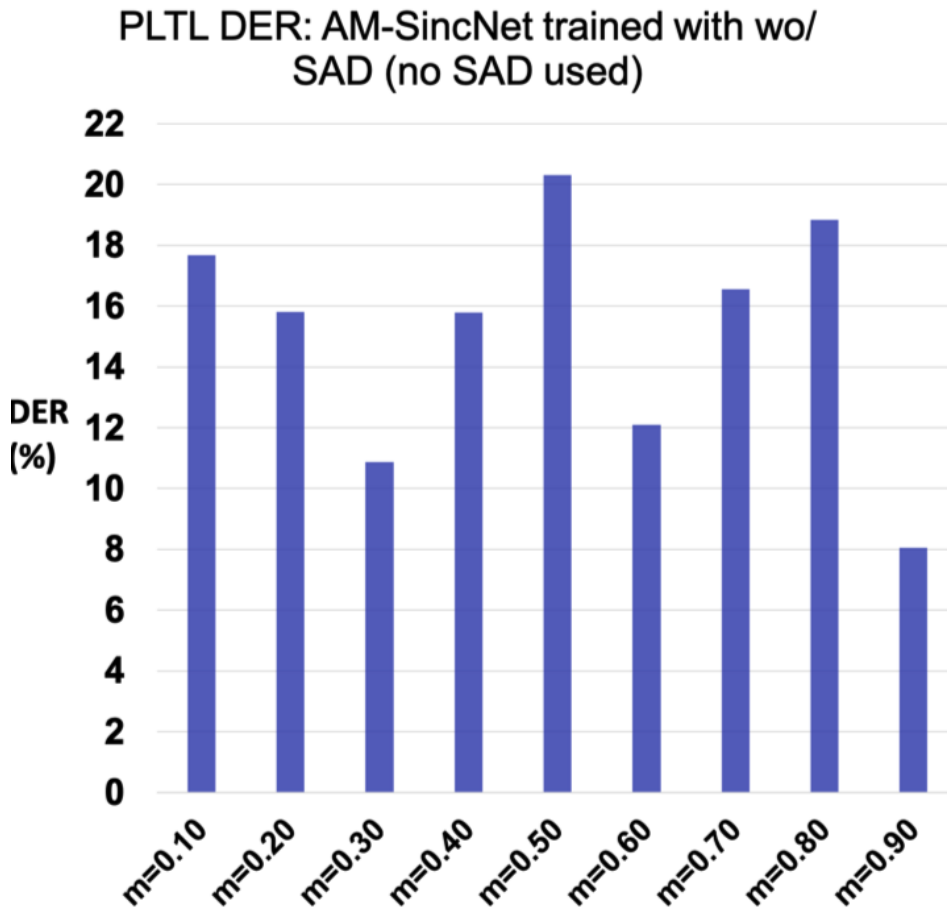


Figure 4.12. Showing the impact of margin parameter used during training AM-SincNet on TIMIT data with no SAD. We used the trained network for extracting speaker embedding from CRSS-PLTL data. These embeddings are used in PLTL diarization pipeline. Best (lowest) PLTL DER is obtained for m=0.90.

or decreasing values of CL parameter (λ) and /or margin parameter (m). Three results from AM-CL-SincNet shows a very little variation in DER unlike significant DER change in corresponding CL-SincNet with same values for CL parameter (λ). Thus, it shows that adding AM-Softmax to CL-SincNet, i.e., AM-CL-SincNet is relatively more robust to changes in CL parameter (λ) values as compared to CL-SincNet.

Fig. 4.14 shows the speaker recognition rate at utterance-level, referred here as sentence error rate (SER) on Librispeech test data when we train STL-SincNet first on TIMIT with ground-truth SAD and then on Librispeech training data without SAD (see Sec. 4.7). We saw

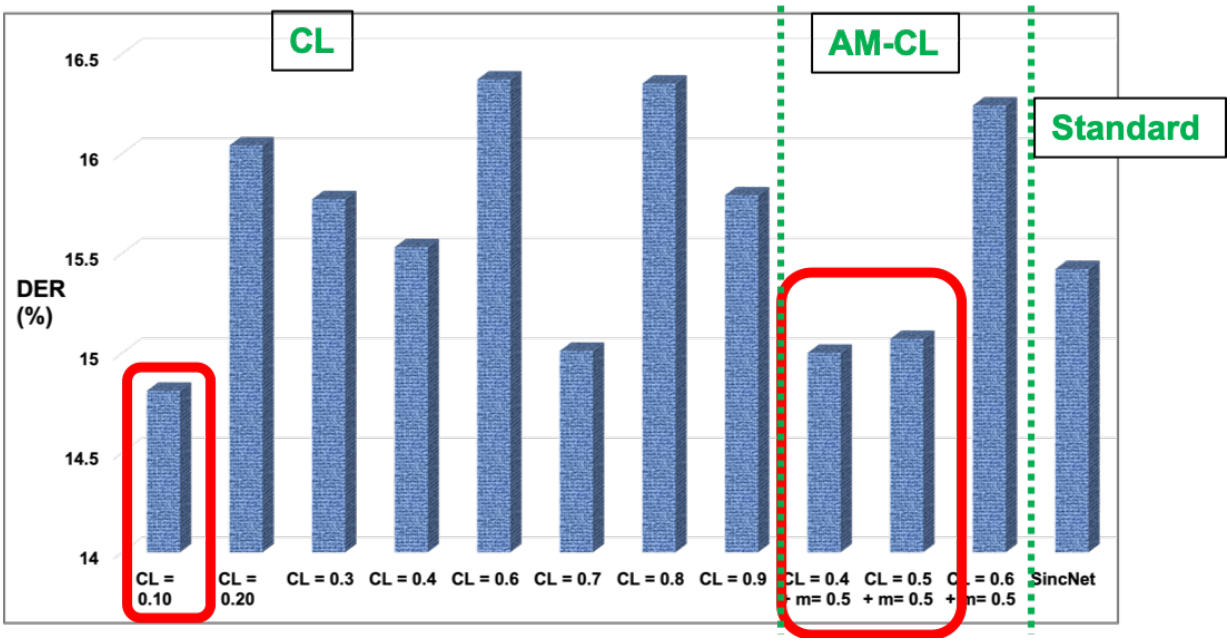


Figure 4.13. Showing the impact of margin parameter in AM-CL-SincNet. There are only a few experiments done to explore the PLTL DER for different parameter set. We can see that margin helps in making the model robust while still preserving the discriminative power of center loss (CL).

that initial performance is significantly better and after 50 epochs it achieved 0.832% error rate which established state-of-the-art on Librispeech speaker recognition. Using standard way for training SincNet model on only Librispeech data converges in 2900 epoch with a final SER of 0.96% (Ravanelli and Bengio, 2018).

Table 4.2. Table summarizing the PLTL DER (%) for all SincNet architectures. It shows only best (lowest) DER from corresponding SincNet architecture. PLTL evaluation data consists of 80 min audio from peer leader in a session.

System	**Best PLTL DER**
Standard SincNet	10.63
AM-SincNet (margin, m = 0.90)	8.04
CL-SincNet	14.81
AM-CL-SincNet (m=0.5, CL parameter, λ=0.5)	15.01
TL-SincNet (F1-avg)	13.29

Table 4.2 shows the PLTL DER (%) for all SincNet architectures. It shows only best (lowest) DER from corresponding SincNet architecture. We used mean normalization of

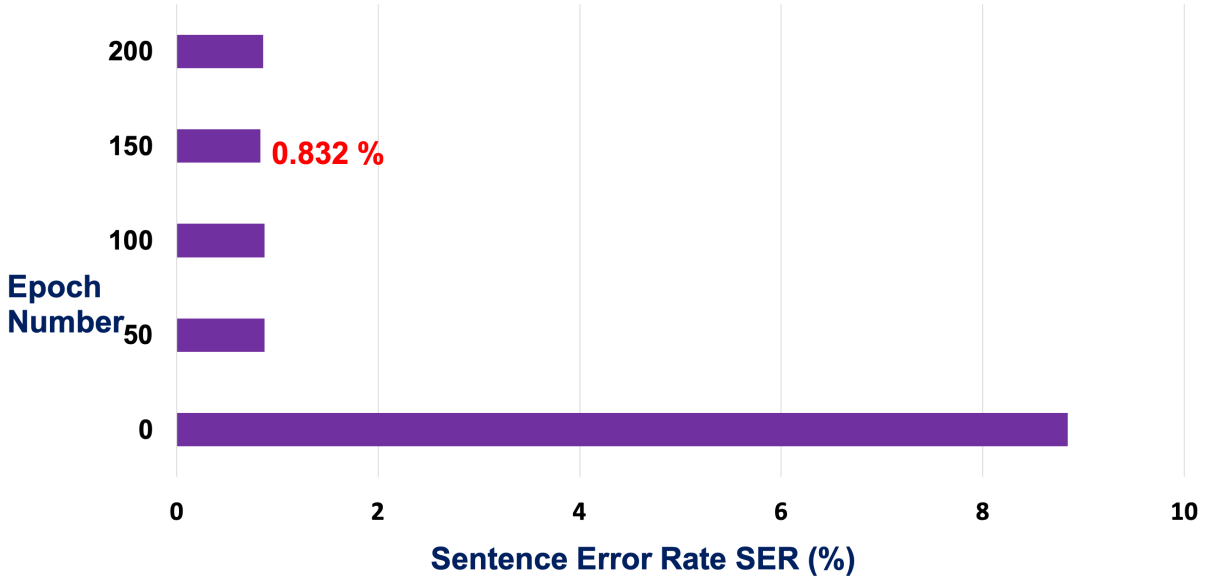


Figure 4.14. Showing the impact of speaker recognition sentence error rate (SER) % for Librispeech test data. STL-SincNet was first trained on TIMIT with ground-truth SAD. Next, the output layer was replaced by two new layers which are learned from Librispeech data. Clearly, the initial performance (epoch 0) is very good. Also, this network converges very fast leading to best SER at epoch 50.

speaker embeddings followed by 100 dimensional PCA for all experiments. Last steps consists of length-normalization and cosine K-means clustering. We see that AM-SincNet gives best results. It is simpler than AM-CL-SincNet and still performs better than AM-CL-SincNet. We were not able to optimize AM-CL-SincNet parameters for best DER as there were many possible experiments. TL-SincNet had only last two layers learned using Librispeech, even if it established state-of-the-art speaker recognition accuracy, it is not suited well for out-of-domain PLTL speaker diarization. A useful next step would be to initialize the whole network from pre-trained SincNet and re-train all layers. This approach is left for future work.

4.9 Summary and Conclusions

In this chapter, we summarize our research efforts using SincNet architectures for speaker diarization and recognition. We proposed four novel architecture based on recently developed

SincNets (Ravanelli and Bengio, 2018) namely: (i) Standard SincNet, (ii) AM-SincNet with AM-Softmax loss, (iii) CL-SincNet with weighted sum of conventional Softmax and center loss (CL), (iv) AM-CL-SincNet with weighted sum of AM-Softmax and CL loss. We leveraged discriminative loss functions such as AM-Softmax and CL-Softmax for enhancing the standard SincNets. We leveraged vanilla transfer learning (VTL) which is a unsupervised transfer learning. In VTL setup, we train the SincNet for frame-level speaker classification using out-of-domain data such as TIMIT or Librispeech corpus. The trained network for referred as SincNet-VTL from which several speaker embeddings were extracted depending on which layer of SincNet was used for embedding extraction. After this, we also leveraged supervised transfer learning (STL) where the network was first trained on TIMIT data using ground-truth SAD and later output layer was replaced by two new layers. Now, the new network was trained for frame-level speaker classification using Librispeech training data. This STL-SincNet established the best speaker recognition accuracy on Librispeech. However, it could not help in enhancing PLTL diarization as we learned only last two layers in new network while freezing the inner layers. We found that AM-CL-SincNet were more robust to change in parameter values as compared to CL-SincNet and AM-SincNet. We did not optimize the parameters of AM-CL-SincNet for PLTL diarization due to many experiments required to do so. We found that AM-SincNet has best DER for margin, m= 0.90. Future research can focus on using more data in STL and parameter optimization in AM-CL-SincNet for enhanced diarization. Overall, the proposed SincNet based speaker modeling has out-performed i-Vector speaker model.

CHAPTER 5

ROBUST SPEAKER CLUSTERING

5.1 Introduction

Speaker Diarization (i.e. determining who spoke and when?) for multi-speaker naturalistic interactions such as Peer-Led Team Learning (PLTL) sessions is a challenging task. Robust speaker clustering plays an important role in diarization pipeline. In this chapter we propose three approaches for speaker clustering namely (i) Mixture of von Mises-Fisher distributions (movMF), (ii) Normalized Fuzzy C-means clustering (NFCM), and (iii) Toeplitz Inverse Covariance-based speaker clustering (TIC). These three methods are benchmarked on CRSS-PLTL corpora and AMI meeting corpus. We also performed another study to understand effect of speech enhancement on speaker clustering performance. Fig. 5.1 shows the proposed

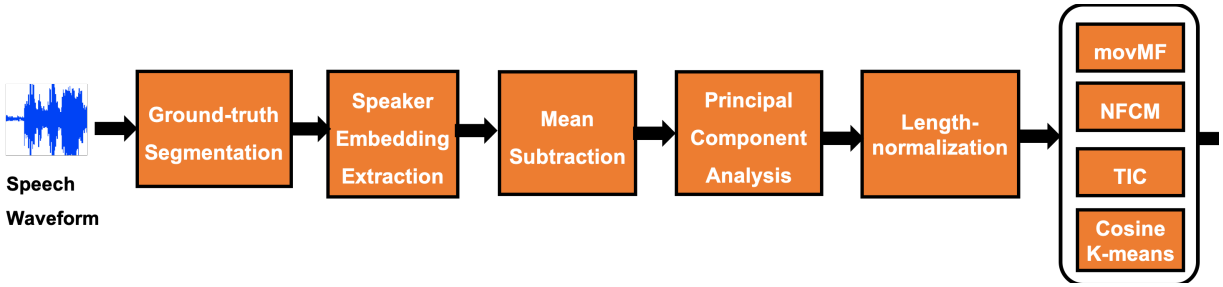


Figure 5.1. Proposed pipeline for evaluation of three proposed clustering approaches and baseline cosine K-means clustering. For all experiments involving clustering studies, we leverage ground-truth speaker segmentation to avoid errors from incorrect segmentation. Three proposed approaches are: (i) movMF; (ii) NFCM and (iii) TIC. Except TIC all algorithms require length-normalization. We perform mean subtraction on each dimensions of speaker features where the mean was computed from entire meeting. PCA was used to study the effects of de-correlating the speaker feature space and dimension reduction on clustering performance. We use DER for performance assessment.

pipeline for benchmarking of proposed speaker clustering approaches. We chose cosine K-means clustering as baseline approach. For all experiments involving clustering studies, we leverage ground-truth speaker segmentation to avoid errors from incorrect segmentation. In

this pipeline, we perform mean subtraction on each dimensions of speaker features where the mean is computed from entire meeting. PCA is used to study the effects of de-correlating the speaker feature space and dimension reduction on clustering performance. We use DER as evaluation metrics for assessment of clustering performance.

The main component of diarization system is speaker clustering. It takes the initial-segments of audio and group it into hard-partitioned clusters. Given the importance of robust clustering for speaker diarization, several approaches were developed such as agglomerative hierarchical clustering (AHC) (Sun et al., 2010), top-down clustering (Meignier et al., 2006), cosine K-means clustering, and HMM-based speaker clustering (Ajmera and Wooters, 2003) etc. Researchers proposed joint speaker segmentation and clustering schemes based on unsupervised analysis (Anguera et al., 2012). In (Zhu et al., 2005), the MAP-adapted Gaussian mixture-models (GMMs) were combined with Bayesian information criterion (BIC) for speaker diarization. A reduced complexity clustering approach leverages modified integer linear programming (ILP) (Dupuy et al., 2014). Recently, speaker diarization based on i-Vectors probabilistic linear discriminant analysis (PLDA) approach was analyzed in details (Salmun et al., 2017). Weighted GMMs were utilized for multi-speaker segmentation for DARPA Hub4 Broadcast News 1997 evaluation (Huang and Hansen, 2006). Unsupervised calibration of PLDA scores was used within i-Vector clustering framework for CALLHOME corpus (Sell and Garcia-Romero, 2014). Previously, von Mises-Fisher distribution were used for text-independent speaker identification based on line spectral frequencies (LSFs) features (Taghia et al., 2013). The square-root of differential LSFs has directional-characteristics, motivating accurate modeling with vMF distributions. The movMF models were used for speaker identification (Taghia et al., 2013), document clustering (Anh et al., 2013), similarity measure for text-snippets (Sahami and Heilman, 2006), clustering gene expression profiles (Dortet-Bernadet and Wicker, 2007), and bio-informatics (Mardia et al., 2007) etc.

Speaker clustering for PLTL sessions is a challenging task due to following reasons: (i) presence of overlapped-speech; (ii) skewed cluster-sizes; (iii) cluster-sizes much smaller than

i-Vector dimension; (iv) significant reverberation and multiple noise-sources etc. Since we input the number of speakers (clusters) in proposed approach, it is referred as informed speaker clustering. Previously, researchers analyzed the length-normalization of i-Vectors concluding that the resultant length-normalized i-Vectors lie on a unit hypersphere (Garcia-Romero and Espy-Wilson, 2011). We model normalized i-Vectors with a mixture of N_c multivariate (d-variate) von Mises-Fisher distributions (movMF) (Banerjee et al., 2005, 2003). Here, N_c is number of speakers and d is the i-Vector dimension. *The normalized i-Vectors lie on a unit hypersphere (Garcia-Romero and Espy-Wilson, 2011) and hence are accurately modeled with a movMF.* The vMF distribution defines a probability density function (PDF) of feature-vectors lying on a unit hypersphere. Modeling the normalized i-Vectors stream from an audio-recording with movMF mixture-model introduces weight parameters, α for each distribution in the mixture. We used the expectation-maximization (EM) algorithm for maximum likelihood estimation (MLE) of movMF model parameters. In this study, we used the EM-based MLE approach developed for movMF mixture-model in (Banerjee et al., 2005). The estimated parameters of N_c vMF distributions of movMF model corresponds to respective speaker clusters.

5.2 MovMF: Mixture of von Mises-Fisher distributions

Speaker clustering for PLTL sessions is a challenging task due to following reasons: (i) presence of overlapped-speech; (ii) skewed cluster-sizes; (iii) cluster-sizes much smaller than i-Vector dimension; (iv) significant reverberation and multiple noise-sources etc. Since we input the number of speakers (clusters) in proposed approach, it is referred as informed speaker clustering. Previously, researchers analyzed the length-normalization of i-Vectors concluding that the resultant length-normalized i-Vectors lie on a unit hypersphere (Garcia-Romero and Espy-Wilson, 2011). We model normalized i-Vectors with a mixture of N_c multivariate (d-variate) vo n Mises-Fisher distributions (movMF) (Banerjee et al., 2005, 2003). Here,

Algorithm 3 Proposed movMF-based Speaker Clustering

Input: (1) Set $\boldsymbol{\chi}$ of n length-normalized i-Vectors of dimensions d from each audio-segment; (2) Number of speakers (clusters), i.e., N_c.

Output: (1) A disjoint-partitioning i.e., clustering of $\boldsymbol{\chi}$ in N_c clusters; (2) Model parameters of N_c d-variate vMF distributions of mixture-model.

METHOD:

Initialize all α_h, $\boldsymbol{\mu_h}$, κ_h for $h = 1, \cdots, N_c$
Repeat
{The hardened Expectation-step of EM}
for $i = 1$ to n **do**
for $h = 1$ to N_c **do**
$f_h(\boldsymbol{\chi_i}|\boldsymbol{\theta_h}) \leftarrow c_d(\kappa_h) e^{\kappa_h \boldsymbol{\mu_h}^T \boldsymbol{\chi_i}}$
end-for
for $h = 1$ to N_c **do**
The hardened-distribution of hidden-variables is given by $q(h|\boldsymbol{\chi_i}, \boldsymbol{\Theta}) \leftarrow \begin{cases} 1, & \text{if } h = \arg\max_{h'} \alpha_{h'} f_{h'}(\boldsymbol{\chi_i}|\boldsymbol{\theta_{h'}}) \\ 0, & \text{otherwise} \end{cases}$
end-for
end-for
{The Maximization-step of EM}
for $h = 1$ to N_c **do**
$\alpha_h \leftarrow \frac{1}{n} \sum_{i=1}^{n} q(h|\boldsymbol{\chi_i}, \boldsymbol{\Theta})$
$\boldsymbol{\mu_h} \leftarrow \sum_{i=1}^{n} \boldsymbol{\chi_i}\ q(h|\boldsymbol{\chi_i}, \boldsymbol{\Theta})$
$\bar{r} \leftarrow \frac{||\boldsymbol{\mu_h}||}{n\alpha_h}$
$\boldsymbol{\mu_h} \leftarrow \frac{\boldsymbol{\mu_h}}{||\boldsymbol{\mu_h}||}$
$\kappa_h \leftarrow \frac{\bar{r}d - \bar{r}^3}{1 - \bar{r}^2}$
end-for
Until *convergence*

N_c is number of speakers and d is the i-Vector dimension. *The normalized i-Vectors lie on a unit hypersphere (Garcia-Romero and Espy-Wilson, 2011) and hence are accurately modeled with a movMF.* The vMF distribution defines a probability density function (PDF) of feature-vectors lying on a unit hypersphere. Modeling the normalized i-Vectors stream from an audio-recording with movMF mixture-model introduces weight parameters, α for each distribution in the mixture. We used the expectation-maximization (EM) algorithm for maximum likelihood estimation (MLE) of movMF model parameters. In this study, we

used the EM-based MLE approach developed for movMF mixture-model in (Banerjee et al., 2005). The estimated parameters of N_c vMF distributions of movMF model corresponds to respective speaker clusters. Recently, we leveraged mixtures of von Mises-Fisher distributions for robust speaker clustering (Dubey, Sangwan, and Hansen, Dubey et al.).

5.2.1 EM-based ML estimation of movMF model parameter

In this section, we present the EM-based ML estimation of model parameters. The d-dimensional normalized i-Vectors are feature-vectors for estimation algorithm. The PDF of a d-variate vMF distribution is given by

$$f(\boldsymbol{x}|\boldsymbol{\mu},\kappa) = c_d(\kappa)e^{\kappa\boldsymbol{\mu}^T\boldsymbol{x}} \tag{5.1}$$

where $||\boldsymbol{\mu}|| = 1$, $\kappa \geq 0$ and $d \geq 2$. Here, the feature-vectors lie on unit hypersphere, i.e., $\boldsymbol{x} \in \mathbb{S}^{d-1}$ and $(\cdot)^T$ denote transpose operation. The normalizing constant, $c_d(\kappa)$ is given by

$$c_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)} \tag{5.2}$$

where $I_r(\cdot)$ is the modified Bessel function of first-kind and order r. The PDF $f(\boldsymbol{x}|\boldsymbol{\mu},\kappa)$ has two parameters, mean direction-vector $\boldsymbol{\mu}$, and concentration parameter κ. The κ indicate how strongly the normalized i-Vectors drawn according to $f(\boldsymbol{x}|\boldsymbol{\mu},\kappa)$ distribution are concentrated along the mean direction-vector $\boldsymbol{\mu}$. Large κ signify substantial concentration along $\boldsymbol{\mu}$. Let us consider a mixture of N_c d-variate vMF distributions as a generative model of recording-specific normalized i-Vectors. Let $f_h(\boldsymbol{x}|\boldsymbol{\theta_h})$ denote the h-th vMF distribution in the movMF model and its parameter-vector is $\boldsymbol{\theta_h} = (\boldsymbol{\mu_h}, \kappa_h)$ for $1 \leq h \leq N_c$. Then, the PDF of movMF mixture-model with N_c component-vMF distributions is given as

$$f(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{h=1}^{N_c} \alpha_h f_h(\boldsymbol{x}|\boldsymbol{\theta_h}) \tag{5.3}$$

where $\boldsymbol{\Theta} = \{\alpha_1, \cdots, \alpha_{N_c}, \boldsymbol{\theta_1}, \cdots, \boldsymbol{\theta_{N_c}}\}$; and α_h are non-negative mixture-weights such that $\sum_{h=1}^{N_c} \alpha_h = 1$. Let $\boldsymbol{\chi} = \{\boldsymbol{\chi_1}, \cdots, \boldsymbol{\chi_n}\}$ be the stream of normalized i-Vectors to be modeled

with mixture-model in Eq. 5.3. Let $\boldsymbol{\zeta} = \{\zeta_1, \cdots, \zeta_n\}$ be the corresponding set of hidden-variables that indicate the component-vMF distribution from which the i-Vectors are sampled. Particularly, $\zeta_i = h$ if $\boldsymbol{\chi_i}$ is sampled from distribution $f_h(\boldsymbol{x}|\boldsymbol{\theta_h})$. In terms of hidden variable-vector $\boldsymbol{\zeta}$, the log-likelihood (LL) of n observed i-Vectors is given by

$$\ln\{P(\boldsymbol{\chi}, \boldsymbol{\zeta}|\boldsymbol{\Theta})\} = \sum_{i=1}^{n} \ln\{\alpha_{\zeta_i} f_{\zeta_i}(\boldsymbol{\chi}_i|\boldsymbol{\theta}_{\zeta_i})\}. \tag{5.4}$$

For a given $(\boldsymbol{\chi}, \boldsymbol{\Theta})$, it is possible to estimate the most likely conditional-distribution of $\boldsymbol{\zeta}|(\boldsymbol{\chi}, \boldsymbol{\Theta})$, and this forms the Expectation-step in EM algorithm. We use the EM algorithm for maximizing the expectation of Eq. 5.4 with constraints $\boldsymbol{\mu_h^T \mu_h} = 1$ and $\kappa_h \geq 0$. As a result, we get the following expressions for movMF model parameters (Banerjee et al., 2005):

$$\alpha_h = \frac{1}{n} \sum_{i=1}^{n} p(h|\boldsymbol{\chi_i}, \boldsymbol{\Theta}), \tag{5.5}$$

$$\boldsymbol{r_h} = \sum_{i=1}^{n} \boldsymbol{\chi_i}\ p(h|\boldsymbol{\chi_i}, \boldsymbol{\Theta}), \tag{5.6}$$

$$\hat{\boldsymbol{\mu}}_h = \frac{\boldsymbol{r_h}}{||\boldsymbol{r_h}||}, \tag{5.7}$$

$$\frac{I_{\frac{d}{2}}(\hat{\kappa}_h)}{I_{\frac{d}{2}-1}(\hat{\kappa}_h)} = \frac{||\boldsymbol{r_h}||}{\sum_{i=1}^{n} p(h|\boldsymbol{\chi_i}, \boldsymbol{\Theta})}. \tag{5.8}$$

The Eqs. 5.7 and 5.8 correspond to Maximization-step in the EM algorithm leading to ML estimates of modal parameters. Given these parameter updates, we now consider update scheme for distribution of $\boldsymbol{\zeta}|(\boldsymbol{\chi}, \boldsymbol{\Theta})$ (i.e., an Expectation-step) to maximize the likelihood of i-Vectors $\boldsymbol{\chi}$ given the estimated parameters from Eqs. 5.7 and 5.8. Using the standard EM algorithm, the distribution of hidden-variables is given by

$$p(h|\boldsymbol{\chi_i}, \boldsymbol{\Theta}) = \frac{\alpha_h f_h(\boldsymbol{\chi_i}|\boldsymbol{\Theta})}{\sum_{l=1}^{N_c} \alpha_l f_l(\boldsymbol{\chi_i}|\boldsymbol{\Theta})}. \tag{5.9}$$

Since computing $\hat{\kappa}$ involved a ratio of Bessel functions (see Eq. 5.8), it is not possible to obtain an analytic solution. Various numerical and/or asymptotic methods are used for

approximating $\hat{\kappa}$. In our study, we used the $\hat{\kappa}$ estimates developed in (Banerjee et al., 2005). Researchers suggested an accurate $\hat{\kappa}$ approximation for hard-partitioned clustering applications (Banerjee et al., 2005). With $A_d(\kappa) = \frac{I_{\frac{d}{2}}(\kappa)}{I_{\frac{d}{2}-1}(\kappa)}$, observe that $A_d(\kappa)$ is a ratio of Bessel functions that differ in their order by one. The $A_d(\kappa)$ can be expressed as a continued fraction given as (Watson, 1995)

$$A_d(\kappa) = \frac{I_{\frac{d}{2}}(\kappa)}{I_{\frac{d}{2}-1}(\kappa)} = \frac{1}{\frac{d}{\kappa} + \frac{1}{\frac{d+2}{\kappa}+\cdots}} \tag{5.10}$$

Letting $A_d(\kappa) = \bar{r}$, we could approximate the above Eq. 5.10 as

$$\frac{1}{\bar{r}} \approx \frac{d}{\kappa} + \bar{r}. \tag{5.11}$$

This leads to following approximation,

$$\kappa \approx \frac{d\bar{r}}{1-\bar{r}^2}. \tag{5.12}$$

Researchers empirically found that the approximation given in Eq. 5.12 could be improved by adding a correction-term $\frac{-\bar{r}^3}{1-\bar{r}^2}$ to it (Banerjee et al., 2005). Thus, the final $\hat{\kappa}$ approximation used in our study becomes

$$\hat{\kappa} = \frac{\bar{r}d - \bar{r}^3}{1-\bar{r}^2}. \tag{5.13}$$

5.2.2 MovMF Speaker Clustering: Algorithm 3

The movMF models were used for speaker identification (Taghia et al., 2013), document clustering (Anh et al., 2013), similarity measure for text-snippets (Sahami and Heilman, 2006), clustering gene expression profiles (Dortet-Bernadet and Wicker, 2007), and bio-informatics (Mardia et al., 2007) etc. We outline the proposed approach in Algorithm 3. It outputs the maximum-likelihood estimates (MLE) of movMF mixture-model parameters. This method essentially iterates over two steps of standard EM algorithm until it converges. In each Expectation-step of EM, i-Vectors are hard-assigned to a single cluster. The *hardened-distribution* of hidden-variables is given in *Step 9* of Algorithm 3. It is important to note

that the denominator in Eq. 5.9 is same for all clusters and hence excluded from "arg max" in *Step 9*. The i-Vectors are hard-assigned to a unique cluster on the basis of derived posterior-distribution. Cluster labels are assigned by computing the arg max of posterior for each i-Vector (*Step 9*).

Next in Maximization-step of EM, the model parameters of component-vMFs are updated using the posteriors of component-vMF distributions given the i-Vectors (*Step 12* to *Step 19* in Algorithm 3). In the proposed hard-assignment approach, the posterior probabilities are restricted to have only binary i.e., 0 or 1 values. With hard-assignments, the distribution of hidden-variables is restricted to assume probability value 1 for some mixture-component and 0 for all others. This hard-assignment strategy maximizes a lower-bound on incomplete log-likelihood (LL) of i-Vectors data (Banerjee et al., 2005). In other words, the expectation over distribution $q(\cdot)$ lower bounds the LL of i-Vectors data.

Upon convergence, the Algorithm 3 output the parameters of mixture-model with N_c component-vMF distributions and the hard-clustering of i-Vectors (*Step 20*). The hard-assignments in *Step 9* reduce the computational-complexity as posterior probabilities are binary values. The proposed clustering approach in Algorithm 3 requires only $\mathcal{O}(N_c)$ computations in each EM iteration. It need to store only the cluster-assignments for all feature-vectors i.e., n integers. These two facts make the proposed clustering approach both computationally-efficient and scalable.

5.3 NFCM: Normalized Fuzzy C-Means Speaker Clustering

Normalized Fuzzy C-Means (NFCM) speaker clustering is an extension of fuzzy c-means (FCM) algorithm for clustering of data lying on unit hypersphere (Kesemen et al., 2016). We propose NFCM as a distribution-free approach for speaker clustering. It is based on the principle that each feature vector can belong to more than one cluster with different membership values in range [0,1] (Bezdek, 2013). NFCM adopts angular difference as the

Algorithm 4 Proposed NFCM Speaker Clustering

Input: GIVEN: (i) n length normalized speaker embeddings extracted from speech segments; (ii) Speaker count, K **Output:** (i) Membership values for all feature vectors for each speaker cluster; (ii) Cluster assignment for each speaker embedding, **spkLabels** = $spkLabels_i$ for i=1, 2,...,n.

METHOD:

1: **Initialize**

2: Choose $m = 2$ and a threshold parameter, $\epsilon > 0$ such as 1e-10 Initialize cluster centers,ϕ_j^0 using uniform distribution between -π and π, i.e, $U(-\pi, \pi)$. Initialize time variable, t=1.

3: Compute angular difference of all feature vectors, θ_i and each cluster center, ϕ_j given by Eq. 5.17, i.e.

4: $\Psi_{ij} = (((\theta_i - \phi_j) + \pi) \text{mod} 2\pi) - \pi$

5: Compute μ_{ij} with Ψ_{ij} as follows:

$$\mu_{ij} = \left(\sum_{k=1}^{K} \left(\frac{||\Psi_{ij}||}{||\Psi_{ik}||} \right)^{\frac{2}{m-1}} \right)^{-1}, \tag{5.14}$$

for i =1, 2,..., N; and j=1,2,...,K.

6: Now, update cluster centers using new values of Ψ_{ij} and μ_{ij}:

$$\phi_j^{(t+1)} = \left(\left(\left(\phi_j^{(t)} + \frac{\sum_{i=1}^{n} \mu_{ij}^m \psi_{ij}}{\sum_{i=1}^{n} \mu_{ij}^m} \right) + \pi \right) \text{mod} 2\pi \right) - \pi \tag{5.15}$$

where j = 1, 2,..., K.

7: Compute difference $||\mu^{(t))} - \mu^{(t-1)}||$.

8: **If** $||\mu^{(t))} - \mu^{(t-1)}|| < \epsilon$, STOP

9: **Else** t = t+1 and obtainO step 3: for computing angular difference.

10: $spkLabels_i$ = arg-max$\{\mu_{ij}\}$ for j=1, 2,..., K.

similarity measure unlike distance-based methods. Use of angular distance make it a consistent and non-parametric approach for directional data. In addition to provide accurate clustering output, NFCM possess low computational time.

Let $\boldsymbol{X = \{x_1, x_2, x3, ..., x_n\}}$ be n speaker embeddings of of dimensions d such that $xi \in \mathbb{R}^d$. Let say there are K speakers, then the clustering approach groups the feature vectors into K groups with cluster centers, $\{\nu_1, \nu_2, ..., \nu_j, .., \nu_K\}$. It works on minimization of

a generalized form of least-squared error functions, as given below:

$$J_m = \sum_{i=1}^{n}\sum_{j=1}^{K} \mu_{ij}^{m}||x - i - \nu j||^2, 1 < m < \infty \tag{5.16}$$

In this study, NFCM is an extension for fuzzy c-means clustering. NFCM leverages angular difference for clustering. The angular difference is defined as

$$\Phi = (((\theta_a - \theta_b) + \pi)\text{mod}2\pi) - \pi \tag{5.17}$$

Lets say, $\Theta_{nfcm} = \{\theta_1, \theta_2, \theta_3, ..., \theta n\}$ be the circular data representing the length normalized speaker embeddings. NFCM tries to minimize following objective function:

$$J_{nfcm} = \sum_{n}^{i=1}\sum_{K}^{j=1} \mu_{ij}^{m}||\theta_i - \Phi_j||^2, 1 < m < \infty \tag{5.18}$$

where m are the weights known as fuzzines parameters. We found $m = 2$ gives the best speaker clustering results. Φ_j is center of the j-th cluster, μ_{ij}^{m} is the membership value of the i-th feature vector assigned to j-th speaker cluster. The membership value, μ_{ij}^{m} must satisfy the following conditions (Bezdek, 2013):

- It lies between 0 and 1 i.e.,

$$\mu_{ij} \in [0, 1] \forall i, j \tag{5.19}$$

- The membership values for each feature vector must sum to one i.e.,

$$\sum_{j=1}^{K} \mu_{ij} = 1 \forall i \tag{5.20}$$

- The sum of all membership values in a cluster must be smaller than the number of data (n) i.e.,

$$0 < \sum_{i=1}^{n} \mu_{ij} < N \forall N \tag{5.21}$$

Algorithm 4 shows the iterations of NFCM that groups n length-normalized speaker embeddings into K speaker clusters. We chose the property of temporal consistency when dealing with short temporal windows.

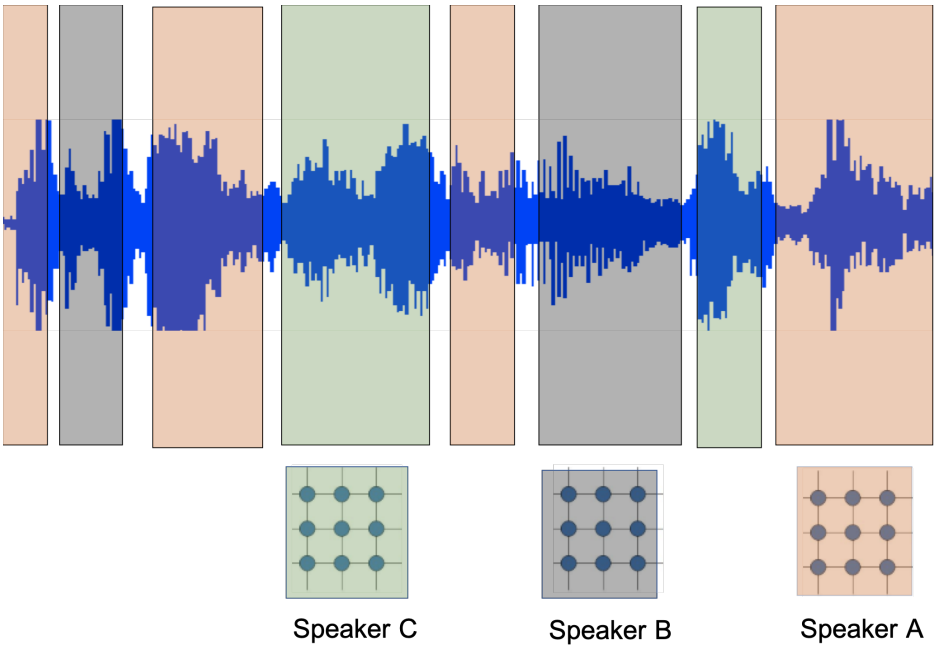


Figure 5.2. The proposed TIC speaker clustering approach takes speaker features extracted from windows of audio signal and perform clustering as a sequence of speaker states. Each cluster A, B or C is characterized by its correlation network defined as a Markov Random Field (MRF). Such MRFs capture the time-invariant partial correlation structure present in all speech segments belonging to that speaker.

5.4 TIC: Toeplitz Inverse Covariance-based Speaker Clustering

Speaker clustering for naturalistic audio such as CRSS-PLTL is a challenging tasks owing to: (i) presence of overlapped-speech; and (ii) significant reverberation and multiple noise-sources. Toeplitz Inverse Covariance (TIC)-based clustering was originally developed for segmenting the real-world time-series such as fitness-tracker, driving data (Hallac et al., 2017). Such real-word temporal data has complicated structure with underlying sequences of few fixed states that repeat in definitive patterns. Robust speaker clustering has a similar property, i.e., there is a small set of speaker (such as 10 for PLTL) that repeat throughout the audio stream within different conversational patterns. Consequently, TIC model satisfies the requirements for speaker clustering using speaker features derived from overlapping speech segments. Each unique speaker is modeled as a correlation network defined by Markov random field (MRF).

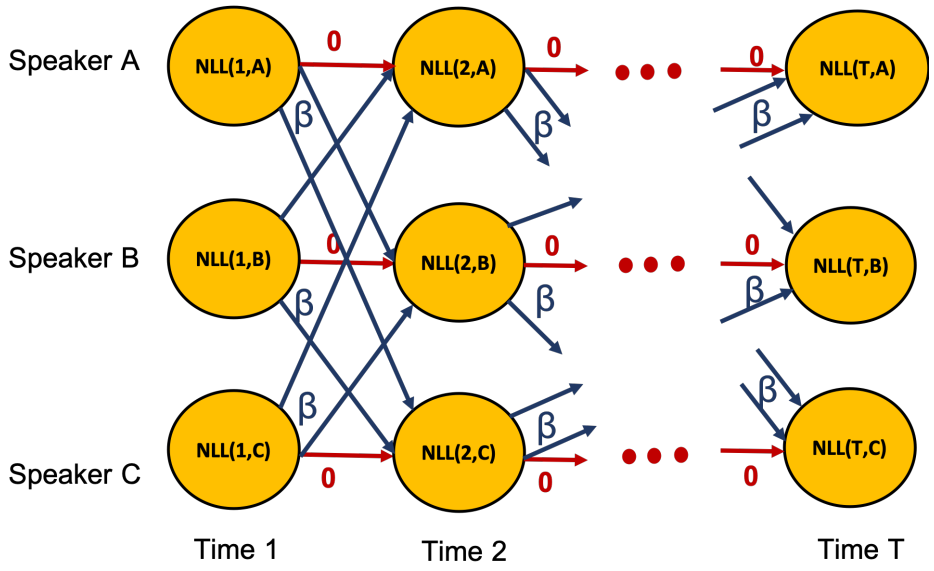


Figure 5.3. The proposed TIC approach takes speaker models from windows of audio signal and perform speaker clustering as a sequence of states. Each speaker cluster, A, B, C is characterized by its Markov random Field (MRF) correlation network. Each speaker MRF captures the time-invariant partial correlation structure of any segments belonging to that speaker.

Each network indicate the interrelationship between different audio segments belonging to the corresponding speaker. We leverage a variant of standard expectation maximization (EM) algorithm for estimating the underlying model and hence cluster the speakers. Speaker models are first initialized. During E-step, feature vectors are assigned to clusters. Next in M-step the model parameters are updated using dynamic programming (DP) and alternating direction method of multipliers (ADMM) (Hallac et al., 2017)

Essentially, this approach views the temporal order of speaker embedding as a sequence of states where each state is characterized by underlying MRF correlation network. Each MRF corresponds to a unique speaker. Unlike centroid-based approaches such as K-means, proposed approach determines the explainable inner structures in feature space belonging to speaker embeddings. This leads to enhanced clustering performance. Each MRF captures time-invariant partial correlation structure in all segments belonging to corresponding speaker.

Algorithm 5 TIC-1: Assign-Clusters

Input: GIVEN $\beta \geq 0$, $-LogL(i,j)$ = negative log-likelihood for i-th feature vector when it is assigned to j-th speaker cluster. K is the number of speakers. Time stamp of i-Vectors (speaker features) is from 1 to T.
Output: FinalPath i.e. cluster assignment for each i-Vector.

METHOD:

Initialize
  previous_cost = list of K zeros
  current_cost = list of K zeros
  previous_path = list of K empty lists
  current_path = list of K empty lists

for i = 1, . . . , T **do**
 for j = 1, . . . , K **do**
  min_index = index of minimum {previous_cost}
  if previous_cost[min_index] + $\beta >$ previous_cost[j] **then**
   current_cost[j] = previous_cost[j] $-LogL(i,j)$
   current_path[j] = previous_path[j].append[j] **else**
   current_cost[j] = previous_cost[minIndex] + β $-LogL(i,j)$
   current_path[j]=previous_path[min_index].append[j]
 previous_cost = current_cost
 previous_path = current_path
final_min_index = index of minimum {current_cost }
FinalPath = current_path[final_min_index]
return FinalPath

5.4.1 Proposed TIC Speaker Clustering

Toeplitz Inverse Covariance (TIC)-based clustering was found to be suitable for segmenting the real-world time-series data such as fitness-tracker and driving data (Hallac et al., 2017). Such temporal data has complicated structure where the underlying sequences of few fixed states repeat in definitive patterns. Robust speaker clustering task possess a similar property, i.e., a small set of speakers (such as 10 for PLTL) that repeat throughout the audio stream in different conversational turns. In this section, we describe the TIC based speaker clustering followed by discussion of experimental results in next section.

Algorithm 6 TIC-2: Proposed TIC-based Speaker Clustering

Input: GIVEN: (i) Algorithm 5 for assigning i-Vectors to speaker clusters; (ii) i-Vectors (features) for time 1 to T.

METHOD:

Initialize
    Speaker cluster parameters, $\mathbf{\Theta}$
    Diarization output, spk_labels = cluster assignment set $\mathbb{C}$
Repeat

    E-step: Assign feature vectors to speaker clusters using Algorithm 5 i.e., map i-Vectors $\Rightarrow$ spk_labels (see section 3.1.)

    M-step: Update speaker cluster (model) parameters by solving the Toeplitz Graphical Lasso (see section 3.2.) i.e., $spk\_labels \Rightarrow \mathbf{\Theta}$

Until Stationarity.
return ($\mathbf{\Theta}$, spk_labels)

In the proposed approach, each unique speaker is represented as a correlation network modeled as a MRF. Such MRF networks capture the interrelationship between different audio segments belonging to the corresponding speaker. A variant of standard expectation maximization (EM) algorithm is leveraged for estimating the underlying models for each speaker and hence grouping the i-Vectors into speaker clusters. First of all, the speaker models are initialized. Next, EM iterations run alternately between Expectation (E-step) and Maximization (M-step). During the E-step, feature vectors are assigned to speaker clusters. Next in the M-step, model parameters are updated using dynamic programming (DP) and alternating direction method of multipliers (ADMM) (Hallac et al., 2017). This process of E-step followed by M-step is repeated until convergence. Essentially, the proposed TIC approach views the temporal order of i-Vectors as a sequence of states characterized by their MRF correlation network. Unlike centroid-based approaches such as cosine K-means, proposed approach models the inner structure present in the i-Vector feature space. This leads to enhanced clustering performance. Each MRF models the time-invariant partial

correlation structure in all feature vectors belonging to the corresponding speaker (Hallac et al., 2017).

5.4.2 E-step: Assign feature vectors to clusters (Algorithm 5)

Given the model parameters (i.e., inverse covariance matrices) Θ_i's for each of the K speakers clusters, solving optimization problem in Eqn 5.26 assigns the T speaker embeddings, $\mathbf{x_1}$, $\mathbf{x_2}$,....., $\mathbf{x_T}$ to these K clusters in such a way that maximizes the likelihood of data while minimizing the number of times the cluster assignment changes across the time. Given K potential cluster assignments of the T points, this combinatorial optimization problem has KT possible mapping of feature vectors to clusters. We assign the cluster using a dynamic programming (DP) approach presented in Algorithm 5 and depicted pictorially in Fig. 5.3. It is equivalent to finding the Viterbi path with minimum cost for the feature sequence of length T.

This task involves fixing the current value of $\mathbf{\Theta}$ and solving the following combinatorial problem for obtaining the cluster assignment set, $\mathbb{C} = \{C_1, C_2, \cdots, C_K\}$:

$$\text{minimize} \sum_{i=1}^{K} \sum_{X_t \in P_i} N(X_t, \Theta_i) + \beta 1_{X_{t-1} \notin C_i} \tag{5.22}$$

This equation assigns each of the T feature vectors to one of the K speaker clusters by jointly maximizing the log likelihood and temporal smoothness ensuring neighboring blocks to be assigned for same speaker. The regularization parameter, β is switching penalty and controls the trade-off between two objectives. For large values of $\beta \to \infty$, all speaker features are grouped together into a single cluster.

5.4.3 M-step: Toeplitz Graphical Lasso (Algorithm 6)

Each speaker cluster is modeled as a Gaussian inverse covariance matrix, $\Theta_i \in \mathbb{R}^{nbXnb}$ where nb is the feature dimension. Since the inverse covariance matrix show the conditional

independence structure between the variables, Θ_i defines a Markov Random Field (MRF) that encodes the structural representation present in all feature vectors of the i-th speaker cluster. Sparse graphical representations prevent overfitting in addition to fetching interpretable results (Lauritzen, 1996). In the M-step of EM algorithm, our objective is to estimate the K inverse covariance matrices, $\mathbf{\Theta} = \{\Theta_1, \Theta_2, \cdots, \Theta_K\}$ using the cluster assignment sets, $\mathbb{C} = \{C_1, C_2, \cdots, C_K\}$, where $C_i \subset \{1, 2, ..., T\}$ obtained from the previous E-step. We can estimate each Θ_i in parallel thus saving execution time. The negative log likelihood of feature vector, X_t to be assigned for i-th cluster with model parameter Θ_i, $N(X_t, \Theta_i)$ can be written in terms of each Θ_i as follows (Hallac et al., 2017):

$$\sum_{X_t \in C_i} N(X_t, \Theta_i) = -|C_i|(\log\det(\Theta_i) + \mathrm{tr}(S_i\Theta_i)) + \gamma \tag{5.23}$$

where $|C_i|$ is the number of feature vectors assigned to the i-th cluster, S_i is the empirical covariance of these feature vectors, and γ is a constant independent of Θ_i. Now, the M-step of EM algorithm becomes

$$\text{minimize} - \log\det(\Theta_i) + \mathrm{tr}(S_i\Theta_i) + \frac{1}{|C_i|}||\lambda \circ \Theta_i||_1$$
$$\text{subject to } \Theta_i \in \mathbb{T} \tag{5.24}$$

where $\mathbb{T}$ is a set of blockwise Toeplitz matrix. Eqn. 5.24 represents a convex optimization problem known as Toeplitz graphical lasso (Hallac et al., 2017). It is a type of graphical lasso problem with additional constraint of block Toeplitz structure on the inverse covariance matrices. Here, λ is a regularization parameter matrix for handling the trade-off between two objectives: (i) maximizing the log likelihood, and (ii) ensuring Θ_i to be sparse. For invertible matrix S_i, likelihood objective lead Θ_i to be near S_i^{-1}. The inverse covariance matrix, Θ_i is constrained to be block Toeplitz and λ is a $nbXnb$ matrix so that it can regularize each sub-block of Θ_i differently. A separate Toeplitz graphical lasso is solved for each speaker cluster at every iteration of the EM algorithm. Since we require several iterations of the

EM algorithm before speaker clustering converges, a fast and efficient approach based on alternating direction method of multipliers (ADMM) is employed for this purpose (Hallac et al., 2017). For more details on solving this Toeplitz graphical lasso, please refer to Section 4.2 in (Hallac et al., 2017). In this section, we propose a model-based speaker clustering approach that explores graphical dependency structures in feature space. We formulate the speaker clustering as a structured inverse covariance estimation problem known as Toeplitz graphical lasso. Our work is motivated by recent success of this approach for unsupervised analysis of physiological data (Hallac et al., 2017).

5.4.4 TIC Clustering as an Optimization Problem

In this section, we will formulate the speaker clustering in terms of an optimization problem. Lets say we have raw i-Vectors from consecutive speech segments denoted as $x_1, x_2, \cdots, x_T$, where $x_i \in \mathbb{R}^d$ and d is the length of speaker features (such as i-Vector). Speaker clustering is the task of distributing these i-Vectors into K groups. We can treat each i-Vector in context of its predecessors or can treat these independently.

Our approach consists of two algorithms: Algorithm 5 - AssignClusters; and Algorithm 6 - Proposed speaker clustering. First algorithm assign each feature vector x_i to a unique cluster. While second one update the model parameters by solving the Toeplitz graphical lasso problem using dynamic programming (DP) and alternating direction method of multipliers (ADMM) approaches. This is similar to standard EM algorithm where these two approaches correspond to E and M steps, respectively. Until convergence, we iteratively cluster the data and then update the model parameters (Hallac et al., 2017).

Lets say, we want to cluster a speech signal, $\mathbf{S_{orig}}$ consisting of T segments that can be of same or different length. Then, the sequences of features for clustering becomes:

$$\mathbf{S_{orig}} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ...\mathbf{x_T}], \tag{5.25}$$

where $\mathbf{x_i} \in R^d$ is d-dimensional speaker model for i-th speech segment. We can either treat each segment independently or in context of its predecessors. Thus, rather than just looking at x_t, we can rather cluster a short sub-sequence of size $b << T$ that ends at time t. Smoothing parameter, β is introduced for ensuring temporal consistency in audio stream. This forces the adjacent segments to be clustered as same speaker. It can be chosen as a regularization parameter according to audio corpus. Model parameters are estimated and iteratively updated by solving Toeplitz graphical lasso. It learns the graphical structure from data based on alternating direction method of multipliers (ADMM). Along with the cluster assignments, this approach estimates the underlying MRF dependency network. Each cluster model is defined by a Gaussian inverse covariance matrix, $\Theta_i \in \mathbb{R}^{dXd}$ where d is the dimensions of input vector. In a study, researchers found that inverse covariance captures the conditional dependency structure between the variables (Koller et al., 2009), hence Θ_i defines a MRF encoding the structural representations in i-th speaker cluster. Being a sparse graphical structure it is robust to over-fitting (Lauritzen, 1996).

The objective is to address for K inverse covariance matrices, $\mathbf{\Theta} = \{\Theta_1, \Theta_2, ..., \Theta_K\}$, one per speaker and find the corresponding cluster assignment sets, $\mathbf{C} = \{C_1, C_{2yyi}, \cdots, C_K\}$ where $C_i \subset \{1, 2, 3, \cdots, T\}$. We assign each of the T points to exactly one cluster. Overall TIC optimization problem is given by

$$\arg \min_{\mathbf{\Theta} \in \mathbb{T}, \mathbf{C}} \sum_{i=1}^{K} \left[||\lambda \circ \Theta_i||_1 + \sum_{X_t \in C_i} \left(N(X_t, \Theta_i) + \beta 1_{X_{t-1} \notin C_i} \right) \right] \tag{5.26}$$

where $N(\cdot)$ is negative log-likelihood. Here, $\mathbb{T}$ refers to a set of symmetric block Toeplitz matrices and $||\lambda \circ \Theta_i||_1$ is an L_1-norm penalty of the Hadamard product to enforce sparse inverse covariance with regularization parameter matrix λ. $N(X_t, \Theta_i)$ denote negative log-likelihood of feature vector X_t to belong in i-th cluster and is given by

$$N(X_t, \Theta_i) = \frac{1}{2}(X_t - \mu_i)^T \Theta_i (X_t - \mu_i) - \frac{1}{2} \log \det(\Theta_i) + \frac{n}{2} \log(2\pi), \tag{5.27}$$

where μ_i is the empirical mean of i-th cluster.

In Eqn. 5.26, β is the smoothness parameter to control temporal consistency, and $1_{X_{t-1} \notin C_i}$ is an indicator function to show if neighboring points belong to the same cluster. Proposed speaker clustering constrains the inverse covariance matrices Θ_i to be block Toeplitz. It means, Θ_i is $nbXnb$ matrix with following structure:

$$\Theta_i = \begin{bmatrix} P^{(0)} & (P^{(1)})^T & (P^{(2)})^T & \cdots & (P^{(b-1)})^T \\ P^{(1)} & P^{(0)} & (P^{(1)})^T & \ddots & \ddots \\ P^{(2)} & P^{(1)} & \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & P^{(1)} & P^{(0)} & (P^{(1)})^T \\ P^{(b-1)} & \cdots & \cdots & P^{(1)} & P^{(0)} \end{bmatrix}$$

where b is window size used in clustering and, $P^{(0)}, P^{(1)}, \cdots, P^{(b-1)} \in \mathbb{R}^{nxn}$. Block $P^{(0)}$ represents the intra-time partial correlations. For MRF corresponding to this cluster, $P^{(0)}$ defines the adjacency matrix of the edges within each layer. The off-diagonal sub-blocks refer to cross-time edges. Essentially, the block Toeplitz structure constraint on inverse covariance matrix ensures time-invariance of speaker models over window size b. In this modeling approach, we identify each speaker cluster by a unique structural pattern in feature space.

Thus, TIC speaker clustering is an optimization problem represented by Eqn. 5.26. It has two variables, cluster assignment sets $\mathbf{C}$ and inverse covariance matrices Θ_i's for each cluster. This optimization problem is highly non-convex and hence no tractable method is available to obtain the globally optimal solution (Hallac et al., 2017). A modified version of standard EM algorithm is used to iterate through two steps in an alternating fashion: (Step 1) assigning features to clusters; (Step 2) Updating model parameters using latest assignments. Even if this approach do not yield globally optimum solution, it was found to work well on clustering

tasks (Hallac et al., 2017; Fraley and Raftery, 2006). Next, we discuss these two steps in detail and present an efficient implementation.

5.4.5 Practical Aspects

Context Dependence

For $\beta = 0$, TIC speaker clustering treats each feature vector independent of each other. Thus, eliminating the temporal consistency constraint and letting each speech segment to be assigned a cluster independent of its location in the audio stream. This is a good setup for speech segments of size 1s or longer. For shorter segments such as 100ms, a small value of β can be a better choice. It provides flexibility to choose the resolution of speaker diarization system.

Regularization

The proposed speaker clustering approach is an optimization problem with two regularization parameters, matrix λ and scaler β. While λ controls the sparsity in MRFs corresponding to each cluster, β governs the temporal smoothness. Higher values of β encourage adjacent feature vectors to be assigned same speaker. In general, λ is a matrix. Practically, it is sufficient to fix it at a constant value for reducing the grid-search efforts. We can also choose prior values of λ and β obtained with help of a development set. Otherwise, these can be a user defined constant values. We choose these values to optimize the DER for each corpus.

Window size

TIC speaker clustering allows us to cluster either a block of several consecutive speaker features or just single speaker features. This provides flexible modeling. Instead of clustering each speaker embedding x_t in isolation, we can also perform clustering over a segment consisting of embeddings from time $t - b + 1$ to t. For this, we concatenate these feature

vectors into a nb dimensional vector, X_t. Here, the window size, b is an important parameter. Proposed clustering approach assumes time-invariant structure for each speaker cluster, thus allowing it to learn cross-time correlations in speaker embedding space.

5.5 Speech Enhancement and Ground-truth Segmentation

Speech enhancement-based diarization pipeline has two stages: (i) ground-truth segmentation; (ii) proposed movMF-based speaker clustering. The ground-truth segmentation information is used for extracting segment-level i-Vectors. We post-process the i-Vectors with principal component analysis (PCA) for dimensionality reduction followed by length-normalization. Length-normalized i-Vectors lie on a unit hypersphere and possess discriminative directional-characteristics. We model the normalized i-Vectors with a movMF mixture-model. Baseline consists of cosine K-means clustering (with cosine distance) for normalized i-Vectors. The evaluation data is derived from: (i) CRSS-PLTL corpus; and (ii) three-meetings subset of AMI corpus. The CRSS-PLTL data contain audio recordings of PLTL sessions which is student-led STEM education paradigm.

In this section, our diarization pipeline consists of three components: (i) speech dereverberation; (ii) ground-truth segmentation; and (iii) proposed movMF-based speaker clustering (or baseline cosine K-means). The purpose of this study is to develop movMF-based speaker clustering so we used ground-truth segmentation information for i-Vector extraction. *Using ground-truth segmentation is important to prevent irrelevant errors due to incorrect segmentation as we focus on speaker clustering.* Previously, researchers found that speaker clustering could be developed independent of other components in diarization pipeline (Sinclair and King, 2013).

The CRSS-PLTL data is significantly reverberated so we perform experiments with both original (raw) and dereverbed audio. We employed weighted prediction error (WPE)-based dereverberation approach developed for REVERB challenge (Yoshioka et al., 2011). After

dereverberation, we get the initial speaker-segments using the ground-truth segmentation information for preventing the irrelevant errors in speaker clustering. We used only raw audio from AMI meeting corpus to avoid reporting too many results.

5.6 Experiments, Results and Discussions

In this section, we discuss the results on CRSS-PLTL and AMI corpora. We first discus the speech enhancement speaker clustering using movMF. Next, we discuss the results with speech enhancement as enhancement has not lead to consistent improvements in DER.

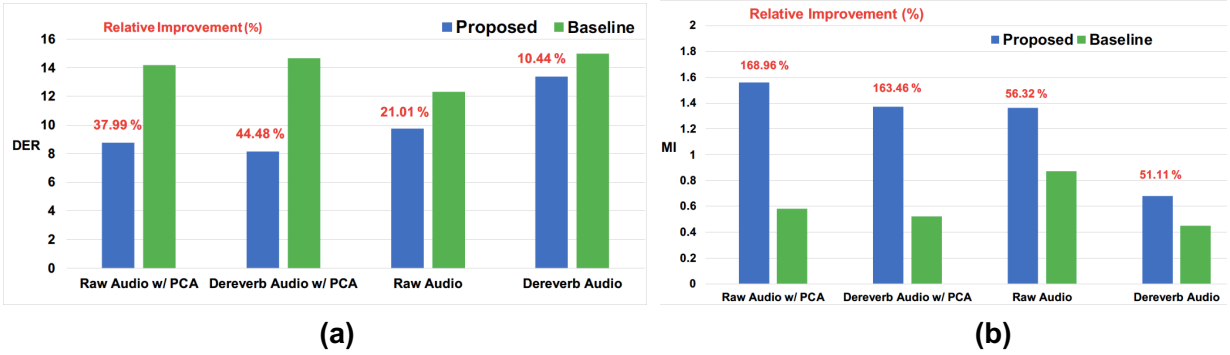


Figure 5.4. PLTL results: (a) Diarization error rate (DER) for proposed and baseline approaches. We used raw audio (original data) and dereverbed audio in our experiments. The "w/ PCA" denotes PCA-based dimension-reduction of i-Vectors with PCA to 51 dimensions before length-normalization. The % relative improvement (reduction) in DER with respect to baseline is shown in red color on top of each bar. The proposed approach is able to significantly reduce the DER elucidating improved performance in all cases. (b) Frame-wise mutual information (MI) for proposed and baseline approaches. The % relative improvement (increase) in MI with respect to baseline is shown in red color above each bar. The proposed approach shows better performance in terms of consistent increase in MI.

5.6.1 Speech Enhancement based movMF Results

Our evaluation PLTL data contains audio of the peer-leader's channel from a 80-minute session with 8 participants. We obtain the ground-truth segmentation and speaker labels from human annotators. Fig. 5.4 compares the performance of movMF approach with baseline on

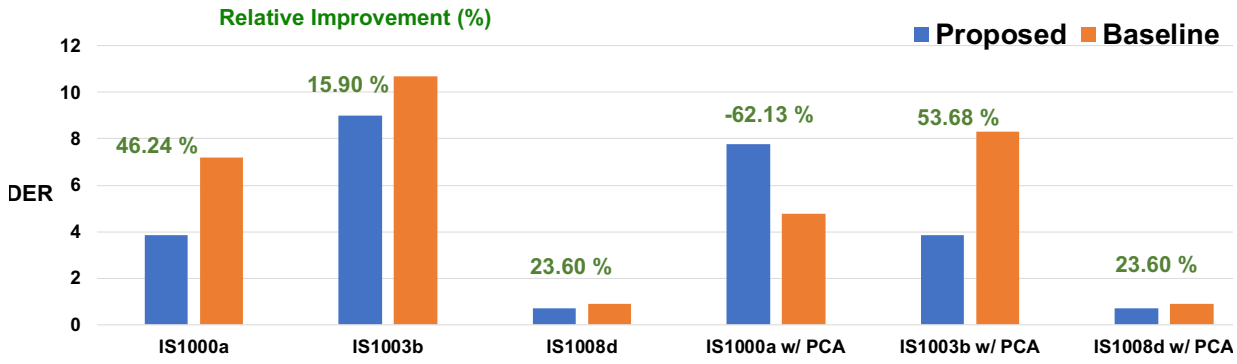


Figure 5.5. AMI (three-meetings subset) results: Diarization error rate (DER) for proposed and baseline approaches. The "w/ PCA" case refers to PCA-based dimensionality reduction of i-Vectors to 65 before length-normalization. The % relative improvement (reduction) in DER with respect to baseline is shown in green color above each bar. The proposed approach showed significant reduction in DER.

PLTL data enhanced with speech enhancement as discussed in Sec. 5.5. Sub-figure Fig. 5.4 (a) illustrates the DER while (b) shows the frame-level MI for raw and dereverbed audio. The majority of PLTL speaker turns were less than one-second (though few lasted over two seconds), we chose 75 (lower) dimensional i-Vectors. We repeated all experiments with PCA for reducing the i-Vector dimension to 51. This PLTL audio contains significantly 10% overlapped-speech that was incorporated as a separate cluster during evaluation. Thus, the number of cluster becomes 9 that includes peer-leader, 7 students and overlapped-speech. Clearly, the proposed movMF approach has improved performance in terms of lower DER and higher MI values compared to baseline. The consistent improvement in all cases with original and enhanced audio, with or without PCA validate the efficacy of movMF model for robust speaker clustering. We have similar observation on three-meetings subset of AMI data as illustrated in Fig. 5.5. We included only DER for AMI data to avoid presenting too many results. The CRSS-PLTL audio has higher levels and more varied forms of distortions as compared to AMI corpus resulting in a challenging diarization scenario. The proposed clustering approach has the ability to adapt the concentration parameter (κ) for each component vMF distribution in the mixture-model. This created a flexible modeling

of normalized i-Vectors that is substantially better than cosine K-means as cosine K-means do not estimate the weight or concentration parameters unlike movMF model. The movMF clustering do a better job by taking advantage of the concentration estimates for each component vMF distribution.

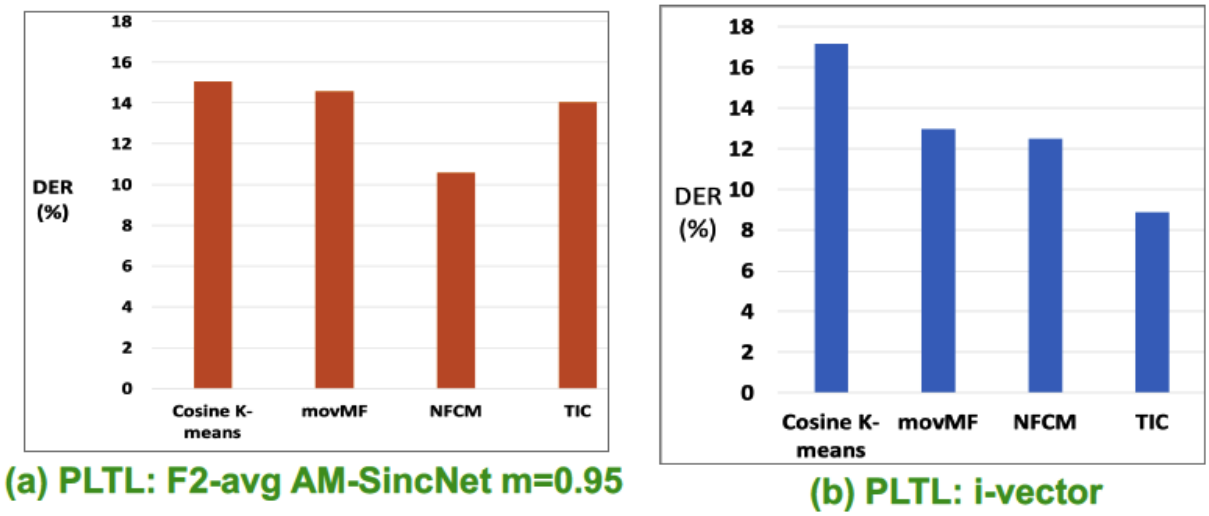


Figure 5.6. PLTL results: (a) Diarization error rate (DER) for three proposed methods (movMF, NFCM and TIC) and baseline cosine K-means clustering. TIC has the best performance without PCA as models the correlation in speaker features. TIC achieves significant reduction in DER as compared to other methods. When SincNet embeddings are used in Cosine K-means, DER significant reduced compared to i-Vectors.

5.6.2 Speaker Clustering Results

We perform comparison of proposed method with cosine K-means clustering on i-Vector and AM-SincNet with 0.95 margin and F2 embeddings (Dubey et al., 2019). Naturalistic audio data from CRSS-PLTL is used during evaluation. PLTL is our target domain so we adopted it for evaluation studies. Fig. 5.6 shows the DER on CRSS-PLTL using proposed clustering. We compare i-Vectors with F2-avg embeddings obtained from a AM-SincNet (m=0.95) model trained on TIMIT data. Unlike NIST RT evaluations (NIST NIST, d), we do not apply any forgiveness collar to the reference human annotations prior to scoring. We ignore the

overlapped-speech for experiments in this section. NIST md-eval scoring script (version-22) was used for DER computations (Wooters and Huijbregts, 2007).

First of all, we can see that all three proposed algorithms are better than cosine K-means baseline. NFCM speaker clustering has best results on AM-SincNet embeddings while TIC has best results when using i-Vector speaker models. We used PCA for cosine K-means, movMF and NFCM experiments. TIC works on learning a correlation model of speaker embeddings hence we do not apply PCA for TIC experiments. Length-normalization is performed for all algorithms discussed in this section. TIC has the best performance without PCA as it works on correlation model of speaker embeddings. TIC achieves significant reduction in DER as compared to other methods. When SincNet embeddings are used in Cosine K-means, DER significant reduced compared to i-Vectors.

Among proposed approaches NFCM has least computational complexity and TIC has the highest. The movMF speaker clustering is moderate complexity and lies between NFCM and TIC. Even if NFCM is simple and fast approach, it led to significant DER reductions while using AM-SincNet speaker embeddings. TIC has best results on i-Vector. It is due to correlation in i-Vector features which is effectively captured by the TIC model.

The PLTL data contains audio of the peer-leader's channel from a 80-minute session with seven students. We obtain the ground-truth segmentation and speaker labels from human annotators. Fig. 5.4 compares the performance of proposed approach with baseline on PLTL data. Sub-figure 5.4 (a) illustrates the DER while (b) shows the frame-level MI for raw and dereverbed audio. The majority of PLTL speaker-turns were less than one-second (though few lasted over two-seconds), we chose 75 (lower) dimensional i-Vectors. We repeated all experiments with PCA for reducing the i-Vector dimension to 51. This PLTL audio contains significantly 10% overlapped-speech that was incorporated as a separate cluster during evaluation. Thus, the number of cluster, N_c= 9 that includes peer-leader, seven students and overlapped-speech. The proposed approach has improved performance in terms

of lower DER and higher MI values compared to baseline. The consistent improvement in all cases with original and enhanced audio, with or without PCA validate the efficacy of movMF model for normalized i-Vectors. We have similar observation on three-meetings subset of AMI data as illustrated in Fig. 5.5. We included only DER for AMI data to avoid presenting too many results. The CRSS-PLTL audio has higher levels and more varied forms of distortions as compared to AMI corpus resulting in a challenging diarization scenario. The proposed movMF speaker clustering has the ability to adapt the concentration parameter κ for each component-vMF distribution in the mixture-model. This created a flexible and accurate modeling of length-normalized i-Vectors that is substantially better than cosine K-means as cosine K-means do not estimate the weight or concentration parameters unlike movMF model. The movMF clustering do a better job by taking advantage of the concentration estimates for each component vMF distribution.

5.7 Summary and Conclusions

This chapter proposed and benchmarked three clustering approaches: (i) movMF, (ii) NFCM, (iii) TIC as compared to cosine K-means clustering. Our evaluation data was derived from naturalistic CRSS-PLTL and AMI meeting corpus. While NFCM is computationally simple, it seems to be effective on AM-SincNet embeddings. NFCM provides soft speaker clustering unlike movMF and TIC. TIC models the speaker features with MRFs and hence capture the underlying correlation model. TIC was best approach for i-Vector features leading to significant DER (%) reductions for CRSS-PLTL data. All the best approaches are better than baseline method on both AM-SincNet speaker embeddings and i-Vector speaker model.

First of all, we can see that all three proposed algorithms are better than cosine K-means baseline. NFCM speaker clustering has best results on AM-SincNet embeddings while TIC has best (least) DER on PLTL data when using i-Vector speaker models. We used PCA for cosine K-means, movMF and NFCM experiments. TIC works on learning a

correlation model of speaker embeddings hence we do not apply PCA for TIC experiments. Length-normalization is performed for all algorithms discussed in this section. TIC has the best performance without PCA as it works on correlation model of speaker embeddings. TIC achieves significant reduction in DER as compared to other methods. When SincNet embeddings are used in Cosine K-means, DER significant reduced compared to i-Vectors.

NFCM has least computational complexity while TIC has the highest. One the other hand, complexity of movMF speaker clustering lies between NFCM and TIC. Simple and fast NFCM algorithm led to significant DER reductions with AM-SincNet speaker embeddings. It is worth noting that Cosine K-means is a special case of proposed movMF-based speaker clustering (Algorithm 3). If we impose all mixture-weights (α_h for $1 \leq h \leq N_c$) to be equal and all concentration parameters (κ_h for $1 \leq h \leq N_c$) to be equal (with any value), then the proposed movMF-based speaker clustering becomes equivalent to Cosine K-means. Thus, we can say that proposed algorithms provide a good variety of supply for speaker clustering ranging from low complexity to high, from correlation model to centroid-based models. When combined with proposed SincNet speaker embeddings and i-Vector speaker models, proposed speaker clustering lead to robust diarization marked with significant reductions in DER for CRSS-PLTL and AMI meeting corpora.

CHAPTER 6

KNOWLEDGE EXTRACTION FOR PLTL INTERACTION ANALYSIS [1]

6.1 Introduction

In this chapter, we summarize our research on audio-based knowledge extraction approaches that is used for analyzing the PLTL interactions. For this purpose, we explore unsupervised analysis and pre-trained models where the models were trained on out-of-domain data. As we described in Chapter 2, PLTL is a student-led STEM education model where a peer-leader facilitate problem-solving in 6-8 students (Carlson et al., 2016; Dubey et al., 2016b). CRSS-PLTL corpora contains multi-stream audio for each session where the number of streams is same as total participants. The salient features of this data are: (i) many segments with overlapped-speech; (ii) short conversational-turns; (iii) multiple noise-sources; and (iv) reverberation. These factors made PLTL speaker diarization challenging. In this chapter, we choose the channel corresponding to PLTL leader for single-channel interaction analysis. Speaker diarization is front-end for interaction analysis of PLTL sessions. Our evaluation set has 8 speakers and lasted for about 80 minutes. It is important to note than many speaker turns lasted for less than 1 second. Education researchers are interested in understanding the impact of participation and group-behavior on class performance. This encouraged us to study audio-based interaction analysis for PLTL sessions. We propose unsupervised dominance score, word-count, question inflection detection, emphasis detection, student participation, and CRSS speech profiler for engagement analysis (Dubey et al., 2016b, 2017). These attributes are estimated using audio signal and diarization output.

Previously, researchers defined behavioral signal processing as computational methods and signal processing approaches for predicting the behavioral patterns in small-group

[1] ©2017 Elsevier Ltd. Portions Adapted, with permission, from H. Dubey, A. Sangwan, J. H. L. Hansen, "H. Dubey, A. Sangwan, and J. H. L. Hansen. "Using speech technology for quantifying behavioral characteristics in peer-led team learning sessions." Computer Speech and Language 46 (2017): 343-366.

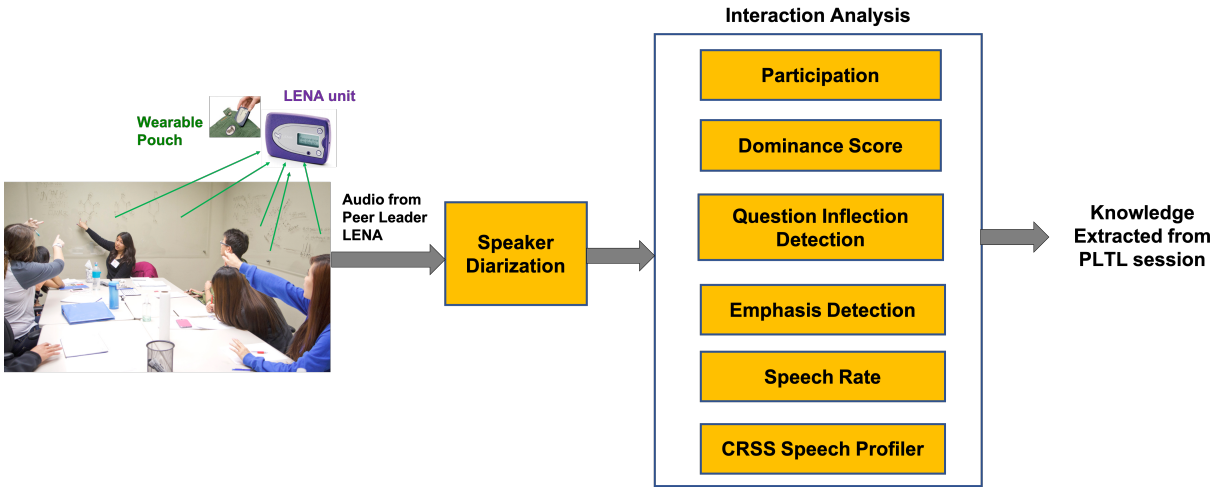


Figure 6.1. Block diagram of proposed speech processing pipeline for analysis of PLTL extraction. We detect several attribute related to PLTL interactions that helps in knowledge extraction for each sessions. Such analysis results are useful for PLTL researchers who wants to study effect of communication behavior on student's performance in semester exams. We leverage diarization output that helps in processing each segment through interaction analysis block.

interactions (Narayanan and Georgiou, 2013). We propose acoustic analyses for extracting features that encapsulate communication behaviors and inter-person turn taking (See Fig. 6.1). Specifically, we extracts following attributes: (1) participation analysis; (2) dominance score; (3) question inflection detection; (4) emphasis detection; (5) speech rate (word count); and (6) CRSS Speech Profiler. These high-level attributes aim to highlight salient aspects of interactions patterns found in PLTL interactions. Fig. 6.1 shows the block diagram of the proposed pipeline that highlights the modular structure of our speech systems.

PLTL audio is first processed with speaker diarization block to know who spoke and when. Second stage consists of speech based methods for extracting attributes related to PLTL interactions. The proposed methods are evaluated on disjoint evaluation datasets taken from CRSS-PLTL corpus (See Table 6.3).

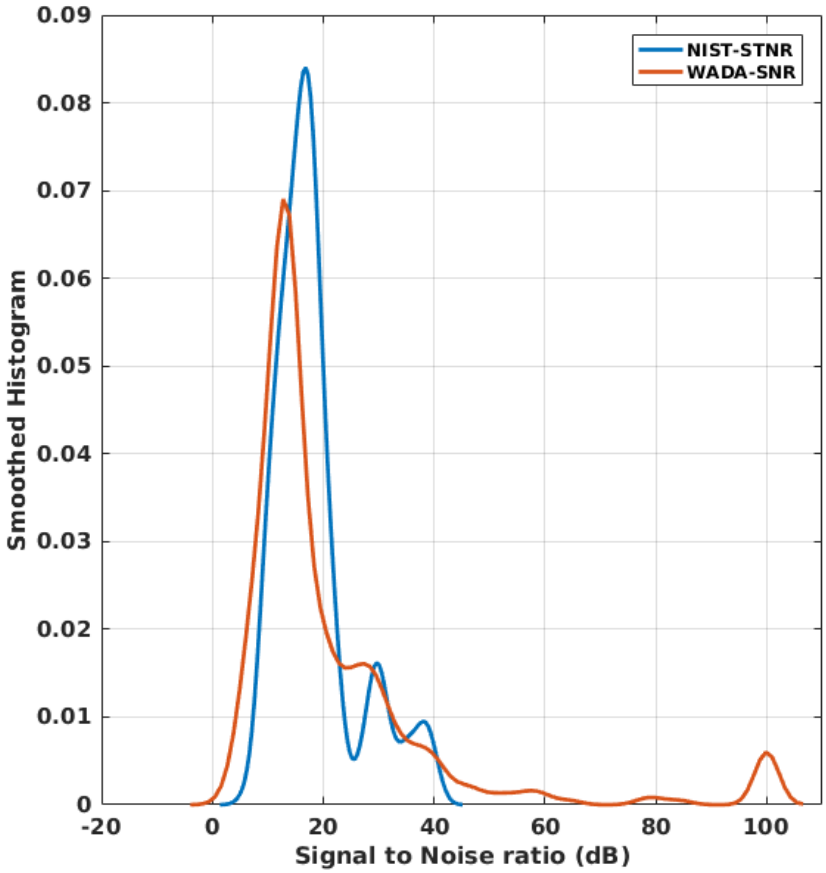


Figure 6.2. The distribution of WADA-SNR (Kim and Stern, 2008) and NIST-STNR (NIST NIST, c) signal to noise ratios(SNRs). Five-minute segments were processed to generated these ratios. We used total of three teams with nine channels each. All teams participated in 80 minute PLTL session, so in total 36 hours of data was used for generating this figure. Since all PLTL sessions were carried out in same space, we could not observe any significant difference in this plot by using more data. NIST STNR have tri-modal distribution while WADA-SNR had bi-modal distributions. We can see that the majority of the segments have SNR between 0 to 15 dB that shows moderate to high noise levels in PLTL data. In addition, huge reverberation is another challenge.

6.2 Exploratory Data Analysis

In this section, we discuss general characteristic of CRSS-PLTL data. Fig. 6.2 shows the distribution of WADA-SNR (Kim and Stern, 2008) and NIST-STNR NIST NIST c signal to noise ratios computed over five-minute segments of 36 hours of PLTL data from three teams. Each team has nine audio streams. The NIST-STNR has tri-modal distribution with

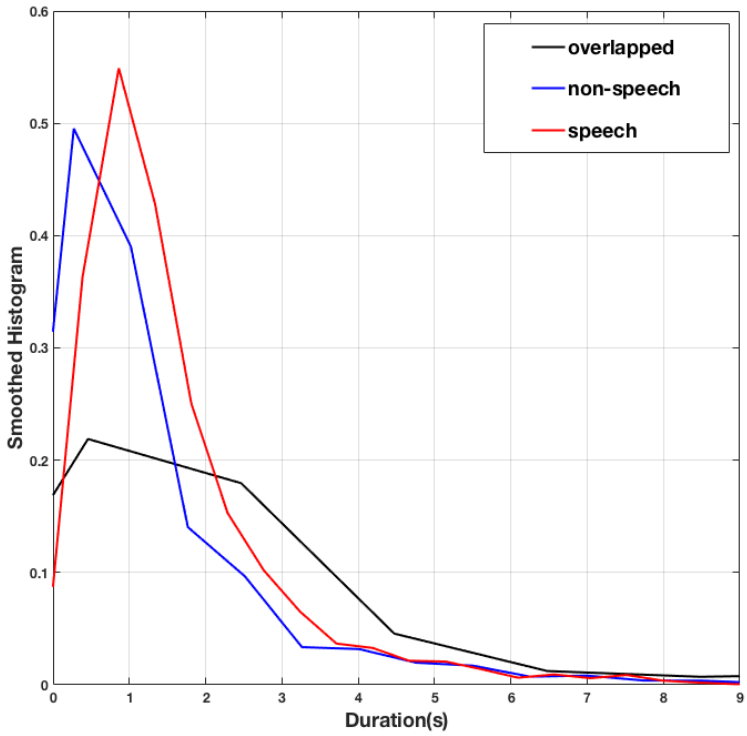


Figure 6.3. Showing distribution of duration of segments with speech, non-speech and overlapped-speech. We could see that most of the segments had duration less than one second. Short-duration segments posed challenge in speaker diarization and behavioral speech processing. The overlapped speech and non-speech accounted for 28.71% and 29.57% of total duration leaving behind only 41.72% speech.

significant first model. One the other had, WADA-SNR has bi-modal distribution where first model is significant. The SNR over five-minute segments was mostly between 0 and 15 dB that showed moderate-to-high noise levels. In addition, huge reverberation was also present that could not be visualized in this plot.

At the end of PLTL sessions, each student and their peer leader completed a form that contained eight behavioral question with four options on Likert-scale (see Fig. 6.4). Questions (Q1, Q2,...,Q8) were given in Table 6.1. These questions belong to three categories, namely PLTL group (PG) assessment, students performance (SP) and overall. The questions Q1, Q2, Q3 and Q4 were regarding the PLTL group (PG) and questions Q5, Q6 and Q7 were based on students performance (SP). The last question, Q8 summarizes the overall assessment. These responses were done on a Likert-scale with four choices, namely strongly disagree(1),

Table 6.1. The questions designed to assess the ground-truth Likert-scale ratings from students. PLTL group (PG) and students performance (SP) refers to two categories of questions developed to assess the student's view on group characteristics and his/her own characteristics, respectively. The Q8 refers to overall assessment.

S.No.	Description	Assessment Type
Q1	My PLTL group was *friendly* today	PG
Q2	My PLTL group was *engaging* today	PG
Q3	My PLTL group was *helpful* today	PG
Q4	My PLTL group was *motivated* today	PG
Q5	I *learned* a lot in today's PLTL session	SP
Q6	I felt *comfortable* with the interaction with my PLTL group today	SP
Q7	My participation in today's PLTL session increased my *confidence* in the course	SP
Q8	Overall, the PLTL sessions are helping me do better in my course	Overall

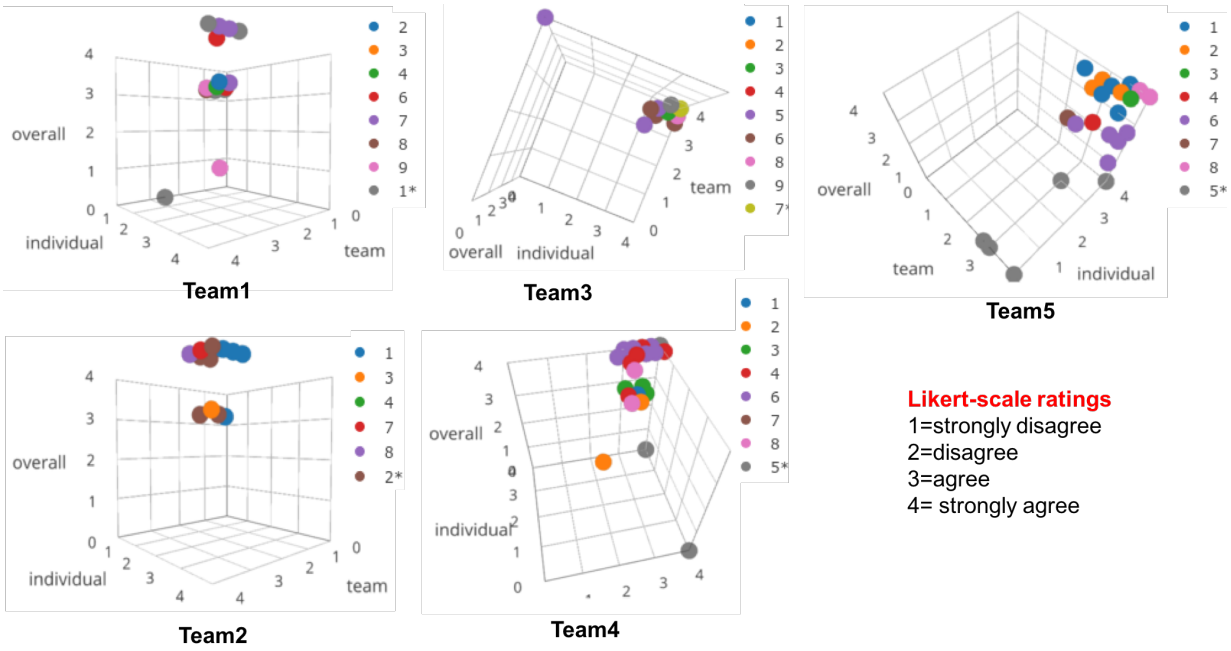


Figure 6.4. Showing overall dynamics of five PLTL teams tracked over eleven weeks in terms of ground-truth Likert-scale ratings obtained from students. These ratings were obtained from feedback forms filled by students after each PLTL session. We discuss more details in Sec. 6.3.

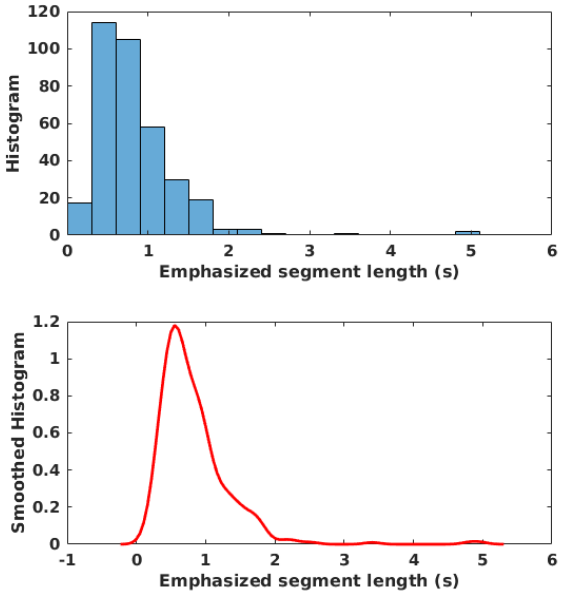


Figure 6.5. Showing distribution of emphasized segment duration for a PLTL session that consisted of approximately 80 minutes audio data. Eight student participated in this session. We could see that most of the emphasized segments have duration less than 1 second.

Table 6.2. Spearman's rank correlation between ground-truth responses of question shown in Table 6.1 for five PLTL groups over 11 sessions for each group, i.e., 55 PLTL sessions in total. We can see high pair-wise correlation in these responses providing hints for combining these into three dimensional scores as shown in Fig. 6.4. We combine PG questions (Q1-Q4) together and SP questions (Q5-Q7) together and left Q8 (overall) as it is. This resulted in three dimensional space for each team that is visualized in Fig. 6.4. The students along with peer leaders are color coded. The peer leaders for each team are marked with asterisk above their numerical index.

S.No.	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**
Q1	1	0.93	0.91	0.90	0.89	0.93	0.89	0.91
Q2	0.93	1	0.92	0.93	0.90	0.92	0.0.90	0.89
Q3	0.91	0.0.93	1	0.90	0.91	0.91	0.91	0.92
Q4	0.90	0.93	0.90	1	0.88	0.91	0.88	0.90
Q5	0.89	0.90	0.92	0.88	1	0.91	0.89	0.91
Q6	0.93	0.92	0.91	0.91	0.91	1	0.92	0.92
Q7	0.89	0.90	0.91	0.88	0.89	0.92	1	0.94
Q8	0.91	0.89	0.92	0.90	0.91	0.92	0.94	1

disagree(2), agree(3), strongly agree(4) (see Fig. 6.4). Each of these eight questions had a response from each student while team leader responded to only PG and overall category of questions. Table 6.1 shows the statement of these questions and its categorization as PLTL group (PG) assessment, students performance (SP) and overall. The Spearman's rank correlation uses ranks instead of the actual values used by the Pearson's correlation. Table 6.2 shows the pair-wise Spearman's rank correlation between ground-truth responses of each question. We could see that among pair-wise correlation between questions Q1 to Q4, the minimum and maximum values were 90% and 93% respectively. The same values for questions Q5, Q6 and Q7 were 89% and 92%. This showed the responses were consistent with respect to categorization. If we see the correlation between Q8 and other questions, we have minimum and maximum values of 89% and 94%. This table gave hints that instead of using responses from eight questions, we could reduce this to a smaller set.

Finally, we averaged the responses to question Q1 to Q4 and called it *team* feature. Similarly, the average of Q5, Q6 and Q7 was called as *individual* feature. The Q8's response was denoted as *overall* feature. We did the averaging operations over all responses from each participant. Fig. 6.4 showed these three features *team*, *individual*, and *overall* for all sessions of each team separately. This serves as visualization of behavioral dynamics of each team. Fig. 6.3 showed the distribution of duration of segments with speech, non-speech and overlapped-speech. We could see that most segments were short with less than 1 second duration. Short-duration segments were challenging with respect to speaker diarization and behavioral speech processing. We used ground-truth information from a PLTL session with approximately 87 minutes of audio data for generating this figure (Eval-Set-7, see Table 6.3). The overlapped-speech and non-speech accounted for 28.71% and 29.57% of total duration leaving behind 41.72% speech. The total number of overlapped-speech, non-speech and speech segments were 205, 738 and 1316 respectively, for this data. The data used for this analysis were Eval-Set-7 as given in Table 6.3. We used this dataset for validating the speech activity

detection based on fusion of DNN-based pitch estimation and TO-combo-SAD (Sadjadi and Hansen, 2013; Ziaei et al., 2014). The results are shown in Table 6.4. Fig. 6.5 shows the distribution of duration of emphasized speech segments. Ground-truth information from a PLTL session with approximately 80 minute duration was used for generating this figure. It showed that most of the emphasized segments had duration less than 1 second.

Table 6.3. Description of evaluation datasets derived from CRSS-PLTL corpus that were used for validating the proposed algorithms. The evaluation datasets were disjoint, i.e., chosen from different PLTL session to avoid bias in annotation process.

Eval dataset	**Duration (minute)**	**Description**
Eval-Set-1	70	Diarization
Eval-Set-2	21	Participation
Eval-Set-3	70	Dominance rating
Eval-Set-4	30	Emphasis
Eval-Set-5	30	Question Inflection Detection
Eval-Set-6	70	Speech rate/ Word count
Eval-Set-7	87	Speech Activity Detection

6.3 Data Preparation and Annotation

Table 6.3 shows the duration and brief description of six evaluation sets that were derived from CRSS-PLTL corpus for experiments discussed in this chapter. These seven evaluation datasets were annotated for diarization ground-truth (Eval-Set-1), speech activity detection (Eval-Set-7), participation analysis (Eval-Set-2), dominance score estimation (Eval-Set-3), emphasis detection (Eval-Set-4), question inflection detection (Eval-Set-5), and speech rate/word count (Eval-Set-6). For CRSS Speech profiler, we used the entire PLTL sessions used for diarization evaluations(Eval-Set-1). The duration of these evaluation sets are provided in Table 6.3. Before conducting the interaction analysis evaluations, human annotator were chosen to perform intelligent listening test for generating the ground-truth. The annotators generated

labels for each attributes. Using different (disjoint) evaluation set was to make sure that annotation bias was the least as some of the studied behavioral characteristics are correlated.

The evaluation set for question inflection detection was 30 minute audio data. We used another 30 minutes of data for emphasis detection. During the listening test, an annotator marked the start-time and end-time of audio segments composed of (1) emphasized-speech regions and (2) interrogative utterances/questions. The goal of this annotation process was to estimate the temporal boundaries of segments with emphasized-speech and question inflection. It is important to note that the semantic aspects were taken into account during ground-truth annotations. For instance, the emphasis was marked on the basis of what was said and how it was spoken in the given context. Same procedure was used for annotating question inflections. However, the algorithms developed for detecting these two phenomenon are based only on acoustics features: (i) fundamental frequency, and (ii) speech energy.

The speaker diarization ground-truth was obtained on Eval-Set-1 with 70 minute audio. Participation refers to annotating the percentage time for which a speaker was active in the PLTL session (Eval-Set-2). For measuring the engagement in terms of speech rate, annotators listened to each five minute segment of PLTL session and note the number of words spoken. Five minutes segment were derived for Eval-Set-6 with 70 minutes audio. Speech activity detection is evaluated on 87 minutes of audio from a PLTL session (Eval-Set-7).

The dominance ratings (ground-truth) were obtained on each five minute segment of Eval-Set-3 (70 minutes). There were seven students in Eval-Set-3. For each five-minute segment, we compute a dominance score (DS) for each of the seven students using unsupervised acoustic analysis explained in Sec. 6.8. Each five-minute segment of Eval-Set-3 was assigning a ground-truth dominance rating (D_{rate}) for each student per segment. Three annotators listened to each five-minute segment and assigned a dominance rating (D_{rate}) for each student per segment. The ground-truth dominance rating, D_{rate}, was a number between 1 and 5. The speakers who were present in the whole session but did not speak in the chosen segment

Table 6.4. Results of SAD systems on PLTL evaluation dataset.

SAD System	**Pfa(%)**	**Pmiss (%)**
(A) TO-Combo-SAD	5.05	12.07
(B) USC-NN-SAD	6.27	15.35
(C) Fusion	8.02	9.42
(D) Fusion + Refinements	10.48	6.65

were assigned a dominance rating, $D_{rate} = 1$. The scores of $D_{rate} = 2$ and $D_{rate} = 5$ were assigned to the least-and most-dominant students who spoked in that segment. For students who spoke in that segment and were neither least-dominant nor most-dominant, we assigned them a D_{rate} between 2.25 and 4.75. It was possible to score 2.25, 2.50, 2.75, 3.0, 3.25, 3.50, 3.75, 4.0, 4.25, 4.50 and 4.75. However, no fractions other than these were used to ensure consistency in evaluations. We averaged the ground-truth rating (D_{rate}) of all three annotator to obtain a final ground-truth that was used for computing the correlation with proposed dominance score (DS).

6.4 Speech Activity Detection for Interaction Analysis

Speech activity detection (SAD) is evaluated on Eval-Set-7 (See Table 6.3 and the data is explained in Sec. 6.2. The evaluation results of SAD algorithms are collected in Table 6.4). Fig. 6.3 shows the distribution of duration of speech, non-speech and overlapped-speech segments. Non-speech often contained several noise sources such as mumbling of far-speakers, writing-on-the-white-board noise (impulsive) in addition to noise from fan and other background sources.

We used DNN-based pitch extractor(see Sec. 6.6) along with TO-combo-SAD (Ziaei et al., 2014) for SAD. The frames that were assigned zero (0 Hz) pitch were declared non-speech. TO-combo-SAD (Sadjadi and Hansen, 2013; Ziaei et al., 2014) was SAD system developed for DARPA RATS data. TO-combo-SAD had shown good performance on naturalistic audio streams such as NASA Apollo mission data. TO-combo-SAD assigned zero (0) for non-speech

and one (1) for speech. We fused the output of both systems for accurate speech activity detection. The frames with non-zero pitch were taken as speech frames and assigned one (1) as SAD output. If both system's output (DNN-based pitch and TO-combo-SAD) were not same, we consider those frames as non-speech. As a results, false alarms were greatly reduced. The non-speech in evaluation dataset has multiple simultaneous sources that results in high false alarm for individual SAD system. We evaluated the SAD system on Eval-Set-7 data as shown in Table 6.4. $Pmiss$ and Pfa refers to miss rate (true-speech detected as non-speech in %) and false alarm rate (true non-speech detected as speech in %), respectively.

In addition to the proposed fused SAD system, we used a supervised SAD system trained on DARPA RATS data (Van Segbroeck, Tsiartas, and Narayanan, Van Segbroeck et al.) and compare its performance with proposed SAD system. The comparison results were shown in Fig. 6.4. This was a supervised Neural Network-based SAD system. The Gammatone, Gabor, long-term spectral variability and voicing features were combined together and used for training the neural network. This system was developed for DARPA Robust Automated Transcription of Speech (RATS) program (Van Segbroeck, Tsiartas, and Narayanan, Van Segbroeck et al.). Researchers extracted features using speech characteristics such as spectral shape, spectro-temporal modulations, periodicity (pitch harmonics), and long-term spectral variability. Researchers used the features from long context-windows to obtain combined feature vector. These features were used for training a neural network (Van Segbroeck, Tsiartas, and Narayanan, Van Segbroeck et al.). The evaluation on DARPA RATS corpora showed accurate results, thus validating the applicability of developed SAD system for highly distorted conditions such as those in DARPA RATS (Van Segbroeck, Tsiartas, and Narayanan, Van Segbroeck et al.).

It is important to note that the PLTL data has (1) not-so-close microphone; and (2) small movement in students, such as moving to white board and writing something, was frequent event that made SAD a challenging task. In addition, huge reverberation and noise corrupted

the speech data further. We would discuss the proposed bottleneck features and informed HMM-based diarization system in Sec. 6.12 and Sec. 6.13.

6.5 Speech Energy

Earlier, we used the formant energy for computing the speaker energy. This energy was leveraged for separating the primary and secondary speakers on each channel of the multi-channel PLTL data (wearer was primary speaker and rest secondary) (Dubey et al., 2016). More often than not, the wearer was assumed to be the closest to their LENA device as compared to other LENA devices. Thus, the audio channel with highest energy could identify the primary speaker. These intensity differences helped in refining diarization output in one of our previous studies (Dubey et al., 2016). Later, we computed the energy of speech signal using wavelet packet decomposition (Dubey et al., 2016a). We choose wavelet packets over formant energy that was used in our earlier studies (Dubey et al., 2016). Formant energy was noise-robust, unlike short-time Fourier transform at the cost of huge computational load. Wavelet packet decomposition was noise-robust and possessed good resolution in time-frequency space with moderate computational load (Wickerhauser, 1991). Wavelet packets provided good time-frequency resolution with reasonable computational expense. The position, scale and frequency parameters characterize the wavelet packets (Wickerhauser, 1991). Traditional wavelet decomposition had only two parameters, namely (1) position; and (2) scale. Wavelet packets could be viewed as a generalized form of wavelet decomposition. Wavelet packets provide better signal resolution in terms of scale, position and frequency dependence. Wavelet packets are bases generated from decomposition of a signal using orthogonal wavelet functions. There are several computationally simple methods for estimating wavelet packets, that made them a better choice for signal decomposition than computationally expensive continuous wavelet transforms.

Traditional wavelet decomposition generates approximation coefficient vector and detailed coefficient vector after first level of decomposition. At next level and successive levels of decomposition, only approximation coefficient vector is re-decomposed into its approximate and detailed components. On the other hand, wavelet packet decomposition allows each detailed coefficient vector to be decomposed in the same way as the approximate coefficient vector (Wickerhauser, 1991). For a speech segment, wavelet packet decomposition generated a complete binary tree allowing a more generic decomposition of the signal. Symlets6 (sym6) wavelet with six levels of decomposition were used for computing the energy. We added the squared wavelet packet coefficient corresponding to the frequency range [50, 2000] Hz for capturing the speech intensity while ignoring the spurious background artifacts and noise. We used the speaker energy for estimating the unsupervised dominance score as discussed in Sec. 6.8 and also in emphasis detection (see Sec. 6.10).

6.6 Robust Pitch Estimation

This section describes the robust pitch extraction using deep neural network trained on stacked spectral features (Pitch Estimation Filter with Amplitude Compression) (Gonzalez and Brookes, 2014). The pitch estimates were later used for detecting curiosity (in terms of question inflection) and emphasized speech. We tested various pitch estimation algorithms such as modified autocorrelation method (De Cheveigné and Kawahara, 2002), Sawtooth Waveform Inspired Pitch Estimator (Camacho and Harris, 2008), Subband Autocorrelation Classification (Lee and Ellis, 2012) and deep neural network (DNN) (Han and Wang, 2014). state-of-the-art pitch tracking method use a deep neural network (DNN) trained on spectral features (Han and Wang, 2014) for predicting the pitch states. DNN-based pitch tracker was the best among four alternatives we tested. The parameters of system used for pitch extraction is given in Table 6.5.

Table 6.5. System parameters for robust pitch extraction method as depicted in Fig. 6.9. The pitch was used for measuring curiosity (in terms of question inflection) and emphasis detection. The super-segments of size 2s were used for detecting emphasis and question inflection.

Parameter	**Value**
Sampling rate	8000Hz
Frame rate	25ms
Skip-rate	10ms
Super-segment size	2s
Features	Pitch Estimation Filter with Amplitude Compression (Gonzalez and Brookes, 2014)
Splicing context (past)	2 frames
Splicing context (future)	2 frames
Number of Hidden Layers in DNN	3
Number of Hidden Nodes (three layers)	1600
Hidden Layer activation	Sigmoid
Output Layer activation	Soft-max
Output Layer dimension (pitch states)	68

We would briefly cover the DNN-based pitch estimator adopted from (Han and Wang, 2014). The stacked spectral features (Pitch Estimation Filter with Amplitude Compression) (Gonzalez and Brookes, 2014) were used to train three-hidden-layer DNN to learn the pitch states. Viterbi decoding was used to connect the probabilistic pitch states, thus fhing the pitch contours. DNN pitch tracker was robust to high amount of noise and worked well for PLTL data. Researchers compared the accuracy of DNN pitch tracker with other methods in (Han and Wang, 2014). Spectral features used for training DNN (See Fig. 6.9) were developed in (Gonzalez and Brookes, 2014). The log-frequency power spectra was normalized to capture long-term information and further filtered to suppress the noise and enhance the harmonic structure in speech frames (Gonzalez and Brookes, 2014).

Pitch Estimation Filter with Amplitude Compression features were earlier used for pitch tracking in noise by peak-picking (Gonzalez and Brookes, 2014). These features were stacked

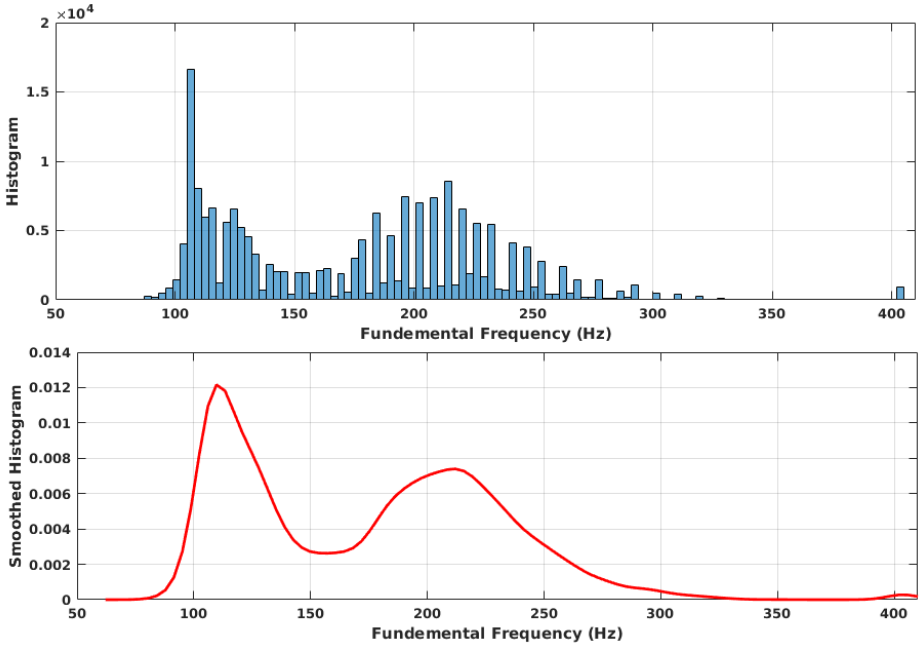


Figure 6.6. Distribution of the fundamental frequency estimates from a PLTL session that consists of eight audio streams of 80 minutes duration. We dropped the non-speech frames (that was assigned a fundamental frequency of zero (0) Hz.

using two past and two future frames as shown in Fig. 6.9 (see Table 6.5). The reverberation and noise in CRSS-PLTL data posed challenge for pitch extraction that necessitated use of DNN-based pitch tracker. We smoothed the DNN extracted pitch using Savitzky-Golay filter (Schafer, 2011) with third order and 11 frames. The smoothing helped in further correction of pitch values for PLTL data. Fig. 6.6 show the distribution of pitch estimates obtained using DNN-based system. It was obtained on a 80 minute audio from a PLTL session. DNN could accurately estimate the pitch eliminating the pitch doubling that was common in unsupervised methods for pitch estimation. The non-speech frames (corresponding to fundamental frequency of 0 Hz) were dropped for plotting this distribution.

6.7 Participation Analysis

Diarization output could be used for extracting participation analysis, that refers to the percentage of total time for which each speaker and their team leader occupied the conversation

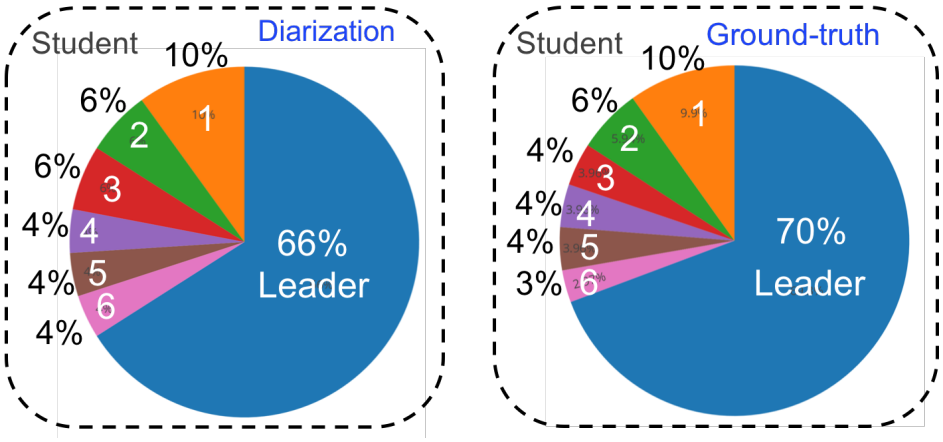


Figure 6.7. Participation analysis of Eval-Set-2 (see Table 6.3) that consisted of 21 minute audio data. It depicts the percentage time for which each individual was speaking. We can see all students occupy comparable fraction of conversation floor while the peer leader occupied the highest fraction.

floor. Fig. 6.7 shows the participation analysis obtained using 21 minutes of data, Eval-Set-2 (See Table 6.3) from a PLTL session. The comparison between diarization-based participation analysis and ground-truth clearly shows that even if diarization error rate is non-zero, we can still derive meaningful participation analysis from it. The percentage values were rounded-up for better visualization (Dubey et al., 2016).

6.8 Proposed Dominance Score [2]

Dominance in human-to-human communication had been studied for several decades (Young, 2016). Dominance is a fundamental aspect of interactions in PLTL sessions. The researchers in social psychology have studied dominance in human interactions (Dunbar and Burgoon, 2005). The speaking time of speakers were found to be correlated with perceived dominance of individuals in groups (Mast, 2002). Researchers in social signal processing studied dominance models developed from multi-modal data. Researchers measured the dominance in meeting

[2] ©2016 IEEE. Portions Adapted, with permission, from H. Dubey, A. Sangwan, and J. H. L. Hansen. "A robust diarization system for measuring dominance in peer-led team learning groups." In 2016 IEEE Spoken Language Technology Workshop (SLT), pp. 319-323, 2016.

using the speaker diarization techniques (Hung et al., 2008). Researchers developed a supervised model for dominance using short-utterances (Basu et al., 2001). However, the model was developed and evaluated on a constrained settings that was very different from real-life situations such as PLTL sessions. Researchers analyzed the interaction between two individuals who debated for 60 seconds. Such controlled settings and short-duration analyses were not applicable for spontaneous conversations such as those in PLTL sessions. Researchers used multi-modal features derived from audio and video streams for analyzing the dominant persons in meetings (Hung et al., 2007).

Researchers used manual transcriptions of meetings for generating semantic metrics that were later used for training static and dynamic models of dominance (Rienks et al., 2006). However, they did not process the audio rather the text was processed to build the supervised models. Such systems could not be deployed for analysis of PLTL groups as they required scripting and training supervised classifiers. Researchers proposed a dominance model for meetings based on supervised learning using multi-modal data (multi-microphone audio and multi-camera video). The audio and visual data were used for training support vector machine classifier. It was used for training the supervised dominance model for meeting conversations (Jayagopi et al., 2009). However, such a system need supervised training on huge amount of labeled multi-modal data and was likely to perform poorly under mismatched conditions. Another limitation was that it could not be used if only audio data were available from PLTL sessions. The proposed dominance score is computed by approach illustrated in Fig. 6.8. We developed an unsupervised feature for measuring dominance (Dubey et al., 2016a). Dominance score (DS) was assigned to each student by unsupervised acoustic analysis of their speech segments. The proposed DS encapsulates the probability of a given student to be dominant in collaborative problem solving. We considered three features derived from speech corresponding to each speaker. This information was available from speaker diarization system. The three features are turn-taken-sum (*turns*) (Larrue and Trognon,

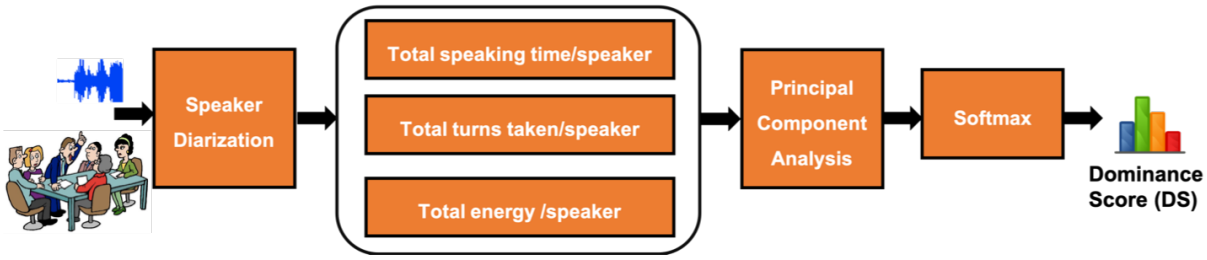


Figure 6.8. Block diagram of proposed speech system for estimating unsupervised dominance score (DS). It uses total speaking time, total turn taken and total energy for each speaker to computed the DS.

1993), speaking-time-sum (*spts*), and speaking-energy-sum (*spens*). These features were motivated from social psychology literature where the dominance of a speaker was found to be correlated with taking more turns in a conversation, speaking for longer duration (Mast, 2002), and with higher energy (Dunbar and Burgoon, 2005). These features were correlated among themselves. For example, a person who was taking many turns was likely to speak for longer duration than others. Also, adding the speaker energy for a longer duration would result in higher *spens*. The turn-taken-sum (*turns*) was the number of turns taken by the speaker in a given session. A conversation turn was decided by a speech segment from the speaker cascaded between that from other speakers and/or speech pauses (non-speech). The speaking-time-sum (*spts*) was the sum of duration of all time-segments (in seconds) belonging to that speaker. The overlapped speech was not taken into account in this sum. Speaking-energy-sum (*spens*) was sum of energies for that speaker's segments.

The speech energy was computed using wavelet packet decomposition (Wickerhauser, 1991) as discussed in Sec. 6.5. The PLTL data had huge reverberation and noise, that necessitated development of better metric for computing speaking energy. We used the Symlets6(sym6) wavelet with six levels of decomposition for computing the speech energy. The coefficients corresponding to frequency range [50, 2000] Hz were summed to obtain the energy. After extracting these three features, *turns*, *spts* and *spens*, we normalized each feature dimension. Let $\mathbf{f}$ be the three dimensional feature-vector, μ and σ being the mean

vector and standard deviation vector. The normalized feature vector, $\overline{\mathbf{f}}$, is given by $\overline{\mathbf{f}} = \frac{\mathbf{f}-\mu}{\sigma}$. Here, the division is point-wise, the mean and variance were calculated over the entire PLTL session (approximately 70-80 minute audio).

We projected these normalized features onto eigen space corresponding to the highest eigen value of the feature space. This was realized by principal component analysis (PCA) that combined the three features into a single feature, named *comb* feature (short form for combined feature). Let us denote the *comb* feature by p. We computed the *comb* feature for each speaker in each segment of the PLTL session. PCA was performed for the whole PLTL session. In this chapter, we divided the entire PLTL session into five-minute segments. A dominance score was estimated for each speaker during five-minute segments.

Lets say, p_i was the *comb* feature corresponding to $i-th$ speaker. For CRSS-PLTL corpus we have six to nine speakers in sessions including team leader. We defined *comb* feature-vector as, $\mathbf{p} = [p_1, p_2, .., p_N]$, where N was the number of speakers. The dominance score (DS) for each speaker was estimated by processing the dominance feature-vector, $\mathbf{p}$, through a soft-max function that convert these numbers into probability scores. Thus, we had

$$DS_i = \frac{e^{p_i}}{\sum_{j=1}^{N} e^{p_j}}, \tag{6.1}$$

for $i = 1, 2, .., N$; where DS_i was the dominance score (DS) of the $i-th$ speaker.

Once we have the dominance score, finding the most and least dominant speaker was trivial. The one with highest score was the most dominant person while the one with lowest was least dominant. In PLTL sessions, the dominance score of each students is an important metric with respect to inter-session variability for all sessions of that team. From previously studied supervised dominance models that predicted only the most dominant speaker, such a comparison would not be possible (Jayagopi et al., 2009; Hung et al., 2007; Huang and Hansen, 2006). Dominance analysis could help in understanding the role of each team member in a PLTL session with respect to learning of their own and others. It could help in choosing

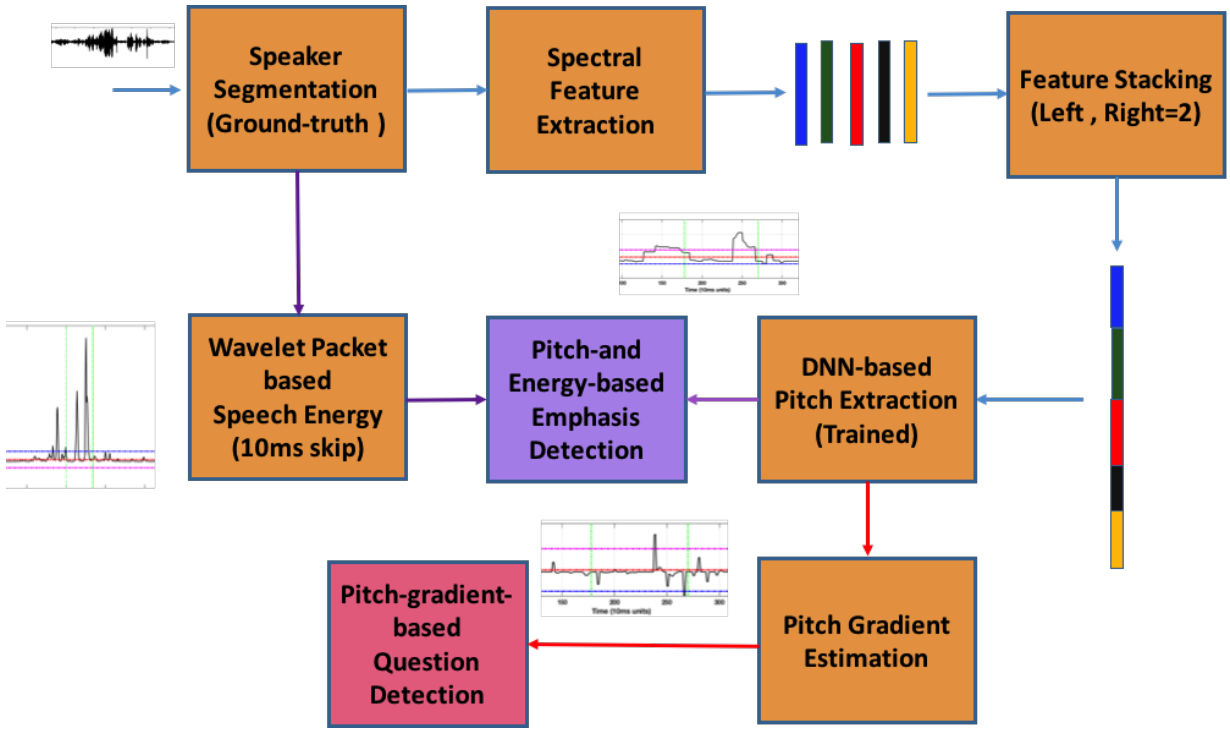


Figure 6.9. Block diagram of the proposed method for detecting question inflections and emphasis in PLTL sessions. Frame-wise pitch was extracted using a deep neural network trained on stacked spectral features (Pitch Estimation Filter with Amplitude Compression) (Han and Wang, 2014). The pitch information along with speech energy was used for detecting the emphasized regions. The pitch gradient was used for detecting the question inflection (a measure of curiosity).

suitable candidates for a PLTL session so as to maximize the learning outcome for each one of them.

Table 6.6. Showing results for emphasis and question-inflection detection. We used the correlation between ground-truth mid-point and point of emphasized speech-region and question-inflection detection. The evaluation used the oracle speaker segments (except the EER calculation) for question-inflection detection.

Quantity	**Correlation**	**root mean squared error(s)**	EER (%)
Question Inflection	0.84	0.51s	12.31
Emphasis	0.78	0.42s	–

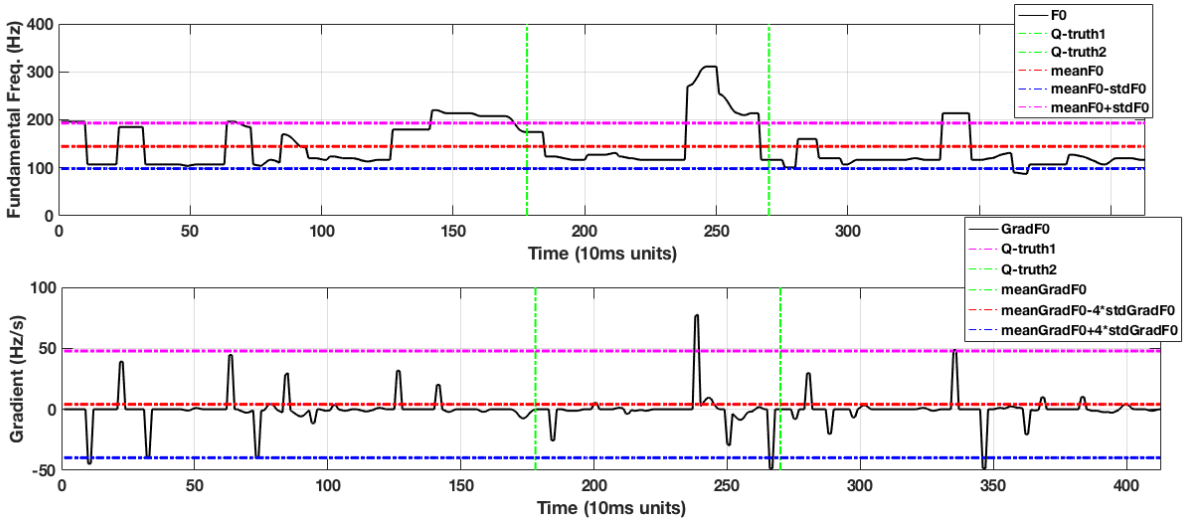


Figure 6.10. Detection of question inflection using gradient of pitch contour. The top sub-figure shows the pitch contour along with start-time (Q-truth1) and end-time (Q-truth2) of the question inflection, and mean-Pitch ± std-Pitch lines. The bottom sub-figure shows the gradient of pitch contour along with mean, and meanGradPitch ± 4*stdGradPitch lines. We could see that question inflection was accompanied by low-to-very high pitch inflation leading to a local maxima at the end of the question (see top sub-figure). We detect the question inflection by a statistical rule as shown in bottom sub-figure. The frames that belong to GradPitch ≥ meanGradPitch ± 4*stdGradPitch corresponds to a question inflection.

6.9 Question Inflection Detection

Curiosity refers to a desire for gaining new information or skill (Renner, 2006). Curiosity was defined in the study as "aurally identifiable trait of the internal desire" of PLTL participants to acquire new information or skills. The curiosity is an important trait in learning (Renner, 2006). A pitch transform was used for detecting the interrogative sentence in (Nagy and Németh, 2016).

Eval-Set-5 (30 minute audio data) was used for evaluating the algorithm for question inflection detection. The audio data was annotated for start-time and end-time of each question. The annotation was done over five minute super-segments. The time-stamps for each question inflection were located. Gradient of the pitch contour for each speaker segment was computed to find the local maximum. Question inflection was detected when the pitch

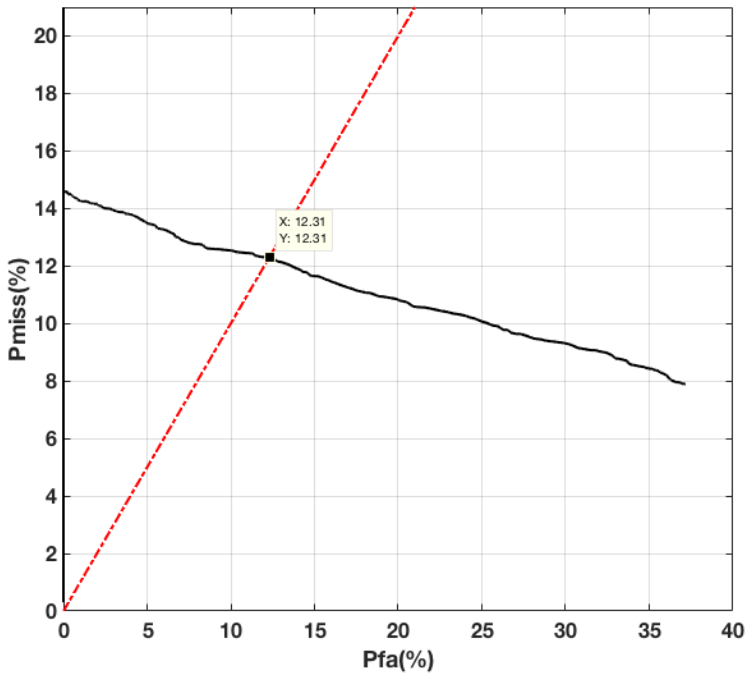


Figure 6.11. Detection error trade-off (DET) curve for 30 minutes of audio data for question inflection detection. The pitch contour from complete signal was mean and variance normalized over non-overlapping 2-second segments. The equal error rate (EER) comes out to be 12.31%. The threshold for detection of question was varied to determine various points (each point corresponds to a miss rate and false alarm rate) shown in this curve.

gradient goes above the value of $meanGradPitch + 4 * stdGradPitch$ (the mean and std are computed using gradient contour over that segment).

We annotated the start-time and end-time of the segment when question was asked. The mid-point of ground-truth question-boundary was used for computing correlation and root mean squared error with algorithm detected question inflection point. Fig. 6.10 shows the pitch variations on a question onset and its neighborhood. It also shows the gradient contour and detection of question inflection. We designed another experiment to study the pitch-based question inflection detection. We took the evaluation audio data and estimated the pitch contour for complete signal regardless of speaker-change boundaries. We performed the mean and variance normalization of pitch contour over each two-second non-overlapping segments. Normalization compensated the long-term effects making the pitch contour robust to acoustic

variability. Normalized pitch was used for detecting the question inflection by choosing a threshold. We varied the threshold from minimum to maximum value (of pitch contour) in small steps. For each threshold values, we get miss probability, $Pmiss$ and probability of false alarm , Pfa (in %) with respect to detection of question inflection.

For EER calculation (DET curve), all frames belonging to the time-interval during which a question was asked, were taken as question inflection points. This is different from root mean squared error and correlation computation where the mid-point of ground-truth question-boundary was compared with point of question inflection detection. Fig. 6.11 shows the detection error trade-off (DET) curve for Eval-Set-5 data. The equal error rate (EER) was 12.31%. Here, $Pmiss$ refers to the frames where we had the questions asked but the system failed to detect it (miss). Pfa refers to the frames where question inflection was falsely detected (false alarm). In this chapter, we used only single channel data for annotation and evaluation for pitch-based question inflection detection for simplicity in evaluation.

Table 6.6 showed the evaluation of question inflection sub-system on PLTL data that consisted of 80 minutes of audio data. The system uses a fixed threshold on speaker-normalized pitch contours for detection of question inflection. The ground-truth question inflection was the end of the question labeled by human annotators. We used a collar that that symmetrically placed around the true question inflection point. If we could successfully detect the question inflection within that collar, it was counted as true detection. We can see using a collar of two seconds, we have a high accuracy given that diarization system was not perfect. Even with non-zero diarization error rate (DER), we get reasonable accuracy over 80% using a collar of 1.5 seconds. With a collar size of one second, we get the accuracy level to 68.42%. The proposed system for question inflection detection was based on the fact that pitch get inflated near the end of a question (also known as question inflection). As pitch ranges vary with speakers, it was crucial to speaker-normalize the pitch contour. Accurate pitch extraction was important to reduce amount of outliers in pitch estimates. The DNN

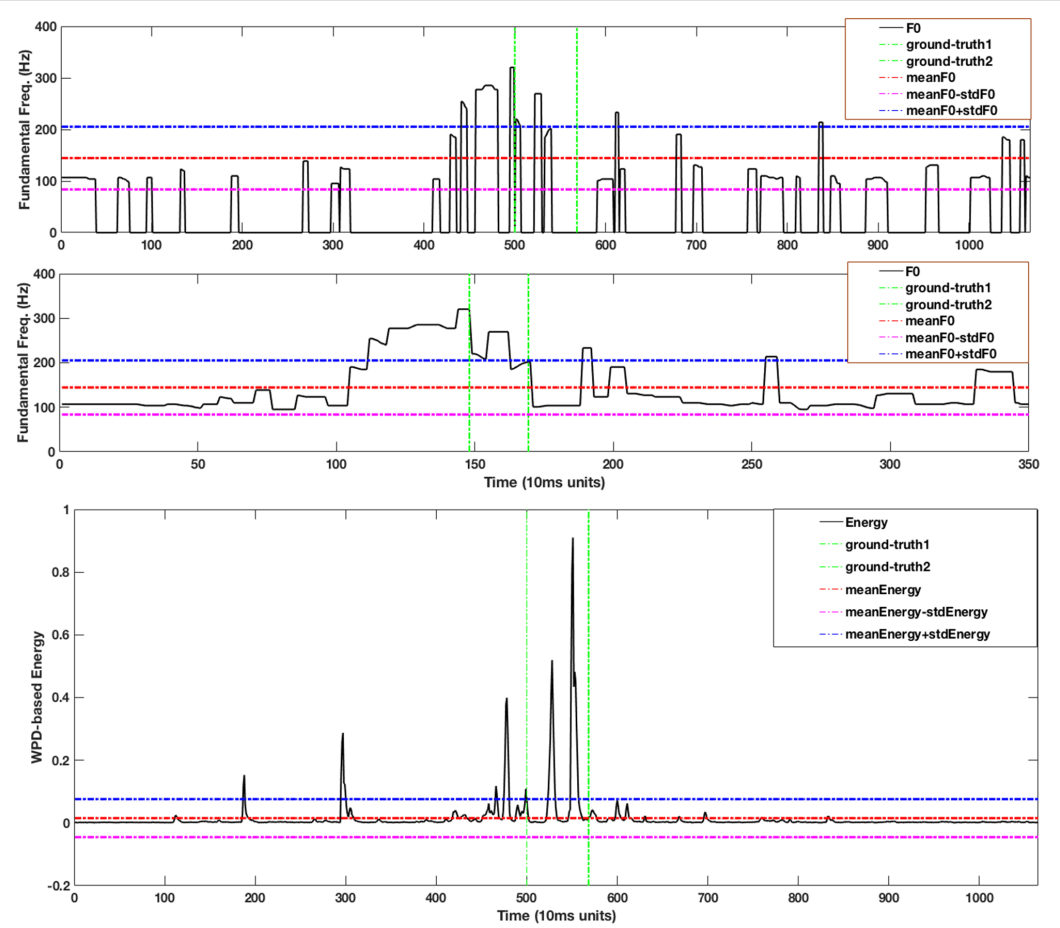


Figure 6.12. Showing pitch-and energy-based emphasis detection. Top sub-figure showed the pitch contour for a speaker segment with emphasized region. The middle sub-figure showed only speech frames (with non-zero pitch). Bottom sub-figure showed the frame-level speech energy obtained using wavelet packet decomposition. When the pitch was higher than $meanPitch + std - Pitch$ and energy was higher that $meanEnergy + stdEnergy$, the emphasized region was detected.

pitch extractor was trained on multi-conditioned TIMIT data containing various noise types added at different SNR levels (Han and Wang, 2014).

6.10 Emphasis Detection

Detection of emphasized speech could help in discovering the "hot-spots" in PLTL sessions wherein important discussions might have happened. Such segments could help education

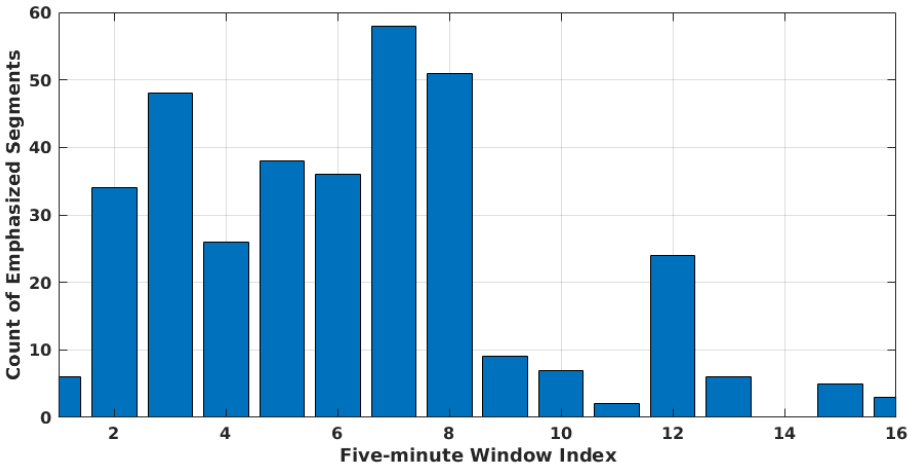


Figure 6.13. We chose a PLTL session with eight students that was organized for 80 minutes. We divide the session into five-minute segments. This bar graph shows the number of emphasized speech regions in each of these five-minute segments. We could be observed that the highest number of emphasized segments occurred around the middle of the session. The last segments were more about logistics and general questions and answers that did not involve emphasized regions.

researchers in understanding and designing the best practices. Student's excitement could be captured by detecting such segments. Emphasized speech regions were important with respect to semantic analysis. Such segments could be further processed with natural language processing (NLP) tools. We have the option of using NLP tools on complete session, however using NLP only on few emphasized segment could reduce computations by eliminating segments that were relatively less important. We used the pitch contour and speech energy for detecting the emphasized speech. These regions identify the important regions in audio data.

The emphasis detection from audio had been studied previously (Chen and Withgott, 1992; Arons, 1994, Arons, 1997). Detecting the emphasized regions helped in quick summarization of spoken documents (Arons, 1997). Such summaries collected the salient features of the recordings and were useful for analysis of technical discussions and daily-life conversations. A HMM-based model trained on huge amount of data was used for emphasis detection in (Chen

and Withgott, 1992). Pitch changes were leveraged for detection of emphasized regions in meetings (Kennedy and Ellis, 2003).

However, the past works (Chen and Withgott, 1992; Arons, 1994, Arons, 1997) had not been tested over long-duration spontaneous speech with several speakers (such as six to eight participants in PLTL session). CRSS-PLTL data had short conversational-turns at several instances in addition to noise and reverberation, thus making the task challenging. Since we estimated the pitch contour and do the analysis for each speaker segment, the pitch range of each speaker is automatically taken into account. As the pitch could change abruptly due to speaker changes (for example, from a male to female speaker), it was important to have accurate speaker segments. The proposed algorithm adapted to the pitch and energy range of a speaker (by operating over non-overlapping two-second windows), and then automatically selected the regions of increased pitch-and energy-activity as a measure of emphasis. Increase pitch and speech energy are markers of an emphasized region while pitch information was found to be more important (Chen and Withgott, 1992).

We proposed detection of emphasized speech using inflated speech energy and increased pitch. The wavelet packet decomposition was used for robust estimation of speech energy as explained in Sec. 6.5. The correlation and root mean squared error between ground-truth (mid-point) and estimated point of emphasis detection were used as figure of merit for this method. Fig. 6.13 showed the distribution of emphasized regions in each five-minute segments of a PLTL session (approximately 80 minutes). We could see the highest number of emphasized speech segment lies in mid of the session. It showed that the "hot-spots" in PLTL sessions were more often during the mid-time.

Fig. 6.12 shows detection of emphasized segments using pitch and energy. Emphasis was detected based on two conditions: 1) energy higher than $meanEnergy + stdEnergy$, and 2) pitch higher than $meanPitch + std - Pitch$. Simultaneous satisfaction of these conditions detected emphasized speech regions. We had the start-time and end-time boundaries for

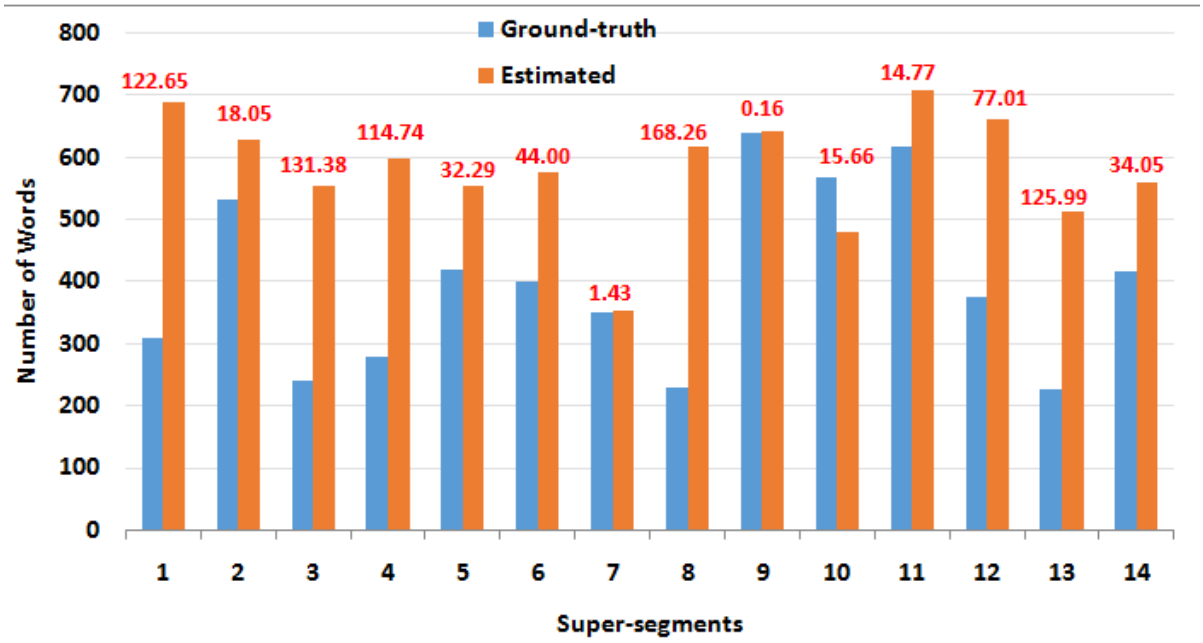


Figure 6.14. The word count ground-truth and estimated using (Ziaei et al., 2016) for Eval-Set-6 (see Table 6.3) that consisted of 70 minutes of audio data. We could see that performance varies from very good to very poor. It depicted the changing acoustic scenarios that affected the quality of the speech signal. The red number above the bars showed the percentage error rate with respect to ground-truth word count. The low errors occur when speaker wore the LENA device and high error occurred due to voice of a distant speaker. The reverberation levels were different for each unique position of the speaker. Very low error in seventh and ninth segment showed that method worked well when speech quality was good and very high errors in first, third and thirteenth segment shows that method proposed in (Ziaei et al., 2016) obtain worse when speaker changes were rapid and/or some of the speakers were far from the LENA device.

emphasized regions from manual annotation as described in Sec. 6.3. We took the mid-point of ground-truth emphasis-boundary and estimated its correlation with algorithm computed point of emphasis detection. Also, we calculated the root mean squared error (in units of second), between these two quantities, i.e., ground-truth and estimated detection point. Table 6.6 shows the evaluation results.

6.11 Speech Rate

Speech rate/word count is an important aspect of vocal communication (Cummins, 2009). Speech rate was useful for quantifying the engagement behavior. Increased speech rate showed more engagement. Researchers used prosodic cues for studying engagement behaviors in children (Gupta et al., 2016). Several interaction scenarios between a child and psychologist were used for validating the developed algorithms. Engagement was predicted using vocal and prosodic cues. Researchers concluded that the engagement information was not only reflected in global cues but also in short-term local cues. Three levels of engagement were used for experimental validation. Fusing global and local cues gave the best results. Even though the experiments were validated in constrained settings, it showed that certain prosodic patterns captured the engagement in dyadic interactions (Gupta et al., 2016).

Several algorithms were developed for estimating the speech rate (Morgan and Fosler-Lussier, 1998; Jiao et al., 2015; Wang and Narayanan, 2007; Ziaei et al., 2016). We benchmarked the method developed in (Ziaei et al., 2016) on Eval-Set-6 (see Table 6.3) derived from the CRSS-PLTL corpus. It consisted of 70 minute audio from a PLTL session.

Fig. 6.14 shows the evaluation of word count algorithm (Ziaei et al., 2016) on Eval-Set-6. We divided the PLTL session into five-minute segments and performed the word count estimation using method proposed in (Ziaei et al., 2016). The red numbers above the bars showed the percentage error rate with respect to ground-truth word count. We could see the performance varying from very low to high error rate. The low errors occurred when speaker wore the LENA device and high error was possibly due to the speech of a distant speaker and rapid short-turns from several speakers (six to eight student were in a PLTL session). The reverberation levels were different for each unique position of speakers. Very low error in seventh and ninth segments showed that method worked well when speech quality was good and very high errors in first, third and thirteenth segment shows that method in (Ziaei et al., 2016) obtain worse when speaker changes were rapid and some of the speaker were far

Table 6.7. The parameters set for proposed diarization system that consisted of three main parts: (1) acoustic feature extraction, (2) stacked autoencoder (autoencoder)-based bottleneck features, and (3) informed HMM-based diarization system.

Parameter	**Value**
Stacked autoencoder input layer dim.	1001
Stacked autoencoder second layer dim.	91
Stacked autoencoder bottleneck layer dim.	21
Number of hidden layers	3
First layer activation	tanh
Hidden layer activation	sigmoid
Initial states in HMM	12-18
Number of GMM components	2-5
Minimum duration of HMM states	0.2s-1s
Splicing context (past)	5 frames
Splicing context (future)	5 frames
Features	MFCC
Window length	25ms
Skip-rate	10ms
Sampling rate	8000Hz

from the LENA device. It showed the necessity to investigate reverberation-and noise-robust methods for speech rate estimation that could work accurately for naturalistic audio streams.

This chapter is a first step towards leveraging speech technology for extracting behavioral characteristics in small-group conversations such as PLTL sessions. Proposed methods were evaluated on CRSS-PLTL corpus. However, these algorithms can be extended to other similar applications such as small-group meetings/conversations. We used robust front-end for speech activity detection (SAD) and speaker diarization. Speech segments from all speaker were later processed with behavioral speech processing block that incorporate several acoustic analyses. Speech algorithms extract features capturing the behavioral characteristics such as participation, dominance, emphasis, curiosity and engagement. Results obtained on CRSS-PLTL corpus using proposed techniques are encouraging and motivate use of behavioral speech processing for understanding practical problems in education, human-to-human communication and small-group conversations.

Table 6.8. Comparison of diarization error rate (DER) for various parameters of the stacked autoencoder-based bottleneck features and informed HMM-based diarization system. I_K is initial number of clusters (hypothesized number of speakers) and I_G is the number of Gaussian components in initial model for over-segmented clusters. All experiments has I_G = 2 and I_K= 12.

SAD	feature dim	$t_{min}(s)$	DER(%)
LIUM			35.80
NO SAD	13-MFCC (* 7= 91 from seven streams)	0.5	41.71
NO SAD	13-MFCC (* 7= 91 from seven streams)	1	33.23
NO SAD	19-autoencoder	0.5	16.64
NO SAD	19-autoencoder	1	**15.83**
Oracle	13-MFCC (* 7= 91, i.e., seven streams)	1	19.98
Oracle	13-MFCC (* 7= 91, i.e., seven streams)	0.5	18.95
Oracle	19-autoencoder	1	**8.05**
Oracle	19-autoencoder	0.5	8.87

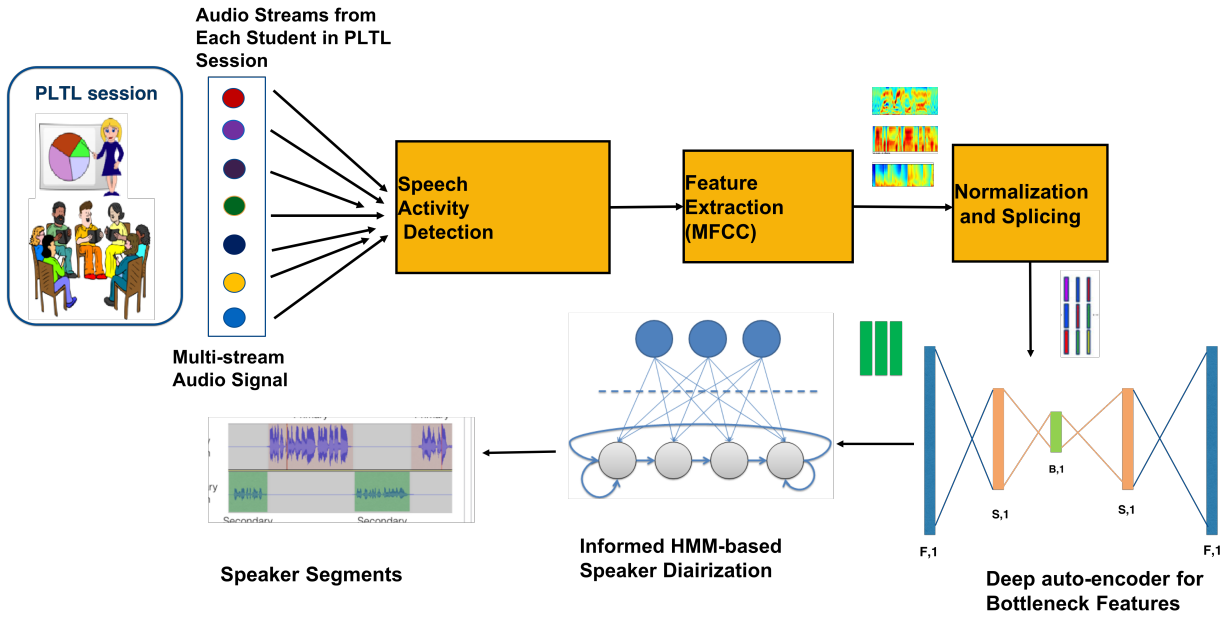


Figure 6.15. Proposed diarization system using autoencoder and HMM. It has two main components: (1) stacked autoencoder based bottleneck features that incorporated splicing with context of five past and future frames and takes acoustic features from all streams of PLTL data; (2) Informed HMM-based diarization system that incorporated the number of students (same as number of audio channels) and minimum duration of conversational-turns as side information.

6.12 Stacked Autoencoder-based Bottleneck Features for Diarization [3]

The proposed scheme is depicted in Fig. 6.15. Deep neural network (DNN) could be used for dimension reduction for high dimensional feature vectors (Hinton and Salakhutdinov, 2006). Autoencoders were found useful in dimension reduction task (Wang et al., 2016). This network was trained in a way that allowed it to learn low-dimensional hidden representation of the data such that taking noisy input, it could reconstruct the input.

Input feature vectors were corrupted with additive random noise. We used 13 dimensional Mel-Frequency Cepstral Coefficients (MFCC). Each feature dimension was mean and variance normalized. We performed splicing of normalized feature vectors by taking five past and future frames. The stacked autoencoder was used for extracting the bottleneck features (bottleneck) from spliced and normalized MFCC features. Several autoencoders were stacked to form a deep network with five layers. Stacked autoencoder was trained using spliced features. Stacked autoencoders were first trained in layer-wise fashion that is a standard way of pre-training. After pre-training, stacked autoencoder was fine-tuned so that it could reconstruct the input features. The input to the stacked autoencoder was corrupted before feeding into it. The reconstruction-loss was minimization criterion for training this network (Vincent et al., 2008).

We used PDNN toolkit (Miao, Miao) with corruption parameter 0.2, learning rate, and momentum factor parameters of 0.01 and 0.05, respectively for realizing the stacked autoencoder. The parameters of the stacked autoencoder used for bottleneck feature extraction was given in Table 6.7. The feature vectors (13-MFCC) were first mean and variance normalized. Let $\mathbf{m}$ was the feature vector, $\mu_{\mathbf{m}}$ and $\sigma_{\mathbf{m}}$ were the mean and standard deviation vectors, respectively. The normalized feature vector, $\bar{\mathbf{m}}$, is given by $\bar{\mathbf{m}} = \frac{\mathbf{m}-\mu\mathbf{m}}{\sigma_{\mathbf{m}}}$.

Since all the channel were delayed and scaled versions of the same speech signal at a given frame, using all channels for diarization was important. Time-spliced feature vectors

[3] ©2016 IEEE. Portions Adapted, with permission, from H. Dubey, A. Sangwan, and J. H. L. Hansen. "A robust diarization system for measuring dominance in peer-led team learning groups." In 2016 IEEE Spoken Language Technology Workshop (SLT), pp. 319-323, 2016.

from each channel were concatenated to form a supervector that consisted of feature vectors corresponding to all PLTL channels. The room where PLTL data was collected has dimensions of 7X10 meters. Thus, the maximum distance between a LENA device and any students (other than the wearer) can be assumed to be within ten meters. Taking the speed of sound in air to be 343 meters per second (m/s), we have the maximum time delay, to be of the order 30ms. This calculation did not accounted for reverberation. We used 25ms windows with 10ms skip-rate for our experiments as given in Table 6.7. We concatenated the features from all streams. The normalized feature super-vectors were spliced by taking five past and future frames. The concatenation was done to incorporate time and intensity differences between various channels of multi-stream PLTL data. The splicing incorporates the long-term context leading to a better quantification of reverberant and noisy speech frames. For a PLTL group with seven streams, the final dimension of spliced features was 11*7*13-MFCC, i.e., 1001.

6.13 Informed-HMM based Diarization System [4]

In this section, we discuss informed-Hidden Markov Model (HMM) for joint speaker segmentation and clustering. HMM system incorporate the bottleneck features from stacked autoencoder system (Gehring et al., 2013) along with two dimensions of side information, *i.e.*, (1) number of speakers; and (2) minimum duration of conversational-turns. Hence, we called the system as informed HMM system. The iterative diarization procedure had three steps: (i) initial segmentation, (ii) merging, and (iii) re-estimation. The diarization for PLTL sessions was different with respect to information available such as speaker-count and turn-statistics. The rapid short-turns, overlapped-speech and significant noise and reverberation made the task challenging. Most of the studied diarization system did not address such challenges (Dubey

[4] ©2016 IEEE. Portions Adapted, with permission, from H. Dubey, A. Sangwan, and J. H. L. Hansen. "A robust diarization system for measuring dominance in peer-led team learning groups." In 2016 IEEE Spoken Language Technology Workshop (SLT), pp. 319-323, 2016.

et al., 2016; Anguera et al., 2012). PLTL sessions had frequent short-segments of size 0.2s to 1s and few segments of size 1-3s. HMMs had been used in previous studies for various audio segmentation tasks in varied forms (Fredouille and Senay, 2006; Madikeri and Bourlard, 2015; Kotti et al., 2008; Ajmera et al., 2002; Huang and Hansen, 2006). However, using side information, application to PLTL and using stacked autoencoder-based bottleneck features were novel contributions with respect to speaker diarization.

Initially, we performed over-segmentation by dividing speech into OS segments where OS was four to six times the expected number of speakers. A HMM with OS states was assumed for initial segments. Each HMM state had an output probability density function that was modeled by M component Gaussian Mixture Model (GMM). Each state of HMM was allowed to have T sub-states to incorporate the minimum duration constraint. All sub-states of a given HMM state (hypothesized speaker cluster) share the GMM corresponding to their state. The HMM system was trained using *Expectation-Maximization* (EM) algorithm. One step aimed to segment the data such that their likelihoods given corresponding GMM parameters were maximized. In next step, the GMM parameters were re-estimated based on new segmentation. Once HMM was trained, we obtained the Viterbi path for each frame. Following it, we used the Viterbi path for checking the binary merging hypothesis based on modified G^3 algorithm (Dubey et al., 2016). After the merge iteration finished, a new HMM with less number of states was trained. The whole process was repeated again till the number of HMM states equaled the number of speakers.

We performed merging based on G^3 algorithm that was a variant of BIC and eliminated the need of the penalty term. The unsupervised G^3 algorithm (Dubey et al., 2016) was used for deciding the binary hypothesis of merging two segments. This trick was first developed to improve the speaker change detection as compared to BIC (Ajmera et al., 2004). In this chapter, we used the same techniques for a different binary hypothesis to decide merging of two over-segmented segments or equivalently two HMM states. There are some modifications

to G^3 algorithm applied for merging most-similar segments (HHM states) at each iteration of the informed HMM-based diarization system. First, the minimum duration of staying in a HMM state was much lower, 0.2s to 0.5s owing to the rapid short conversational-turns. The initial segments were modeled with a Gaussian Mixture Model (GMM) with only M_s components. After merging two initial segments modeled with M_s components, the merged segment was modeled with $2M_s$ components. Thus, the number of parameters in GMM model for merged segment is same as the sum of number of parameters in child segments. Consequently, the number of parameters remains the same at each merging step, and hence the penalty term in BIC criterion (See Equation 6.5) is eliminated.

Let $\mathbf{X_m} = [\mathbf{X_1}, \mathbf{X_2}]$ be the feature matrix corresponding to the merged HMM states. Merging two segments, $\mathbf{X_1}$ and $\mathbf{X_2}$ into $\mathbf{X_m}$ can be formulated as the following binary hypothesis: $\mathcal{H}_0$ *vs.* $\mathcal{H}_m$, where $\mathcal{H}_m$ denotes merging, and $\mathcal{H}_0$ denotes no merging. To facilitate the test, we build models for both hypotheses. GMMs were used to model $\mathbf{X_1}$, $\mathbf{X_2}$ and merged segment $\mathbf{X_m}$. Let ψ_{X_m} be the parameter vector of the GMM with $M_s = M_1 + M_2$ component estimated for the merged segment, $\mathbf{X_m}$. Let, ψ_{X_1} and ψ_{X_2} be the parameter vector of the GMMs with M_1 and M_2 components, estimated for the child segments, $\mathbf{X_1}$ and $\mathbf{X_2}$, respectively. Under the assumption of independence and identical distribution of feature vectors in segments $\mathbf{X_1}$ and $\mathbf{X_2}$, we can represent the log likelihood $\mathcal{L}_{\mathcal{H}_0}$ and $\mathcal{L}_{\mathcal{H}_m}$ for hypotheses $\mathcal{H}_0$ and $\mathcal{H}_m$, respectively as

$$\mathcal{L}_{\mathcal{H}_m} = \log(p(\mathbf{X_1}|\psi_{X_m})) + \log(p(\mathbf{X_2}|\psi_{X_m})), \tag{6.2}$$

$$\mathcal{L}_{\mathcal{H}_0} = \log(p(\mathbf{X_1}|\psi_{X_1})) + \log(p(\mathbf{X_2}|\psi_{X_2})), \tag{6.3}$$

where $p(\mathbf{X_m}|\psi_{X_m})$ is the likelihood of merged segment, X_m given the model, ψ_{X_m}, and so on. The merging decision is made based on the $D_{merging}$, defined as

$$D_{merging} = \mathcal{L}_{\mathcal{H}_m} - \mathcal{L}_{\mathcal{H}_0}, \tag{6.4}$$

However, if we used Bayesian Information Criterion (BIC) for making the merging decision, then corresponding to Equation 6.4, we have following expression for BIC merging:

$$D_{BIC} = \mathcal{L}_{\mathcal{H}_m} - \mathcal{L}_{\mathcal{H}_0} - \frac{1}{2}\nu\Delta \log N_m, \tag{6.5}$$

where ν is a constant usually assigned a value of 1.0 and N_m is the number of feature vectors in merged segment, $\mathbf{X_m}$. Here, Δ is the difference in number of parameters in merged model, ψ_{X_m} and sum of parameters in child models, ψ_{X_1} and ψ_{X_2}. All segments were evaluated for $D_{merging}$. The segments with $D_{merging} \geq 0$ were merged. Once the merging done, the new HMM of smaller size was estimated where the GMM for each state was re-estimated using the EM algorithm. The acoustic features belonging to that HMM state (cluster) were used to re-estimate the corresponding GMM.

The parameters of proposed diarization system was shown in Table 6.7. The results of diarization system were given in Table 6.8. It is important to note that MFCC features from all the seven streams were used in HMM-based diarization for comparing its performance with bottleneck features. We could see that bottleneck feature captured useful statistics of multi-stream audio data that resulted in better accuracy using informed HMM-based diarization system.

We extracted 13-dimensional MFCC features from each of the seven streams of the PLTL session. After concatenating the features from each stream we get a feature super-vector of dimensions 91 (=13*7). After splicing the feature super-vectors with five past and future frames (see Fig. 6.15), we get the final dimension of features as 1001 (=11*91). Spliced feature super-vector was fed to a stacked autoencoder for extracting the bottleneck features of dimension 21. Stacked autoencoder with three hidden layers was chosen where the middle hidden layer acted as bottleneck layer. The bottleneck features were fed to the informed HMM-based diarization system. We used the Oracle SAD in the proposed system to validate the accuracy of HMM-based joint segmentation and clustering. However, we performed

another case-study by formulating non-speech as an additional HMM state. We compared the diarization accuracy of bottleneck features (derived from raw MFCC features form each of the seven steams) and raw acoustic features (13-MFCC from each of the seven streams). Thus, the concatenation of MFCC features from multi-stream was done in both cases ensuring that it was a fair comparison between two approaches (raw features and bottleneck).

Table 6.8 showed the diarization accuracy in various cases. The "NO SAD" case refers to not using any SAD labels and modeling non-speech as an additional HMM state. We knew that the non-speech has several distinct varieties, such as silences (with extreme noise of varied types), overlapped speech etc. This made the diarization, a challenging task without SAD labels. It led to degradation in diarization accuracy (see Table 6.8). We could see that the bottleneck features combined with HMM was robust with respect to change in minimum duration constraints and to some extent is robust to absence of SAD labels. state-of-the-art LIUM baseline (Meignier and Merlin, 2010) was borrowed from our earlier work for comparison (Dubey et al., 2016). We could see an absolute improvement of approximately 27% in terms of DER over the baseline LIUM system and approximately 12% improvement was due to bottleneck features instead of using raw MFCC features (Oracle SAD, one second time-constraint).

6.14 CRSS Speech Profiler for Engagement Detection

Fig. 6.16 shows the CRSS speech pipeline for knowledge extraction in PLTL team learning. This system is modular with three stages namely: Front-end, Feature extraction, Analysis backend. The proposed speech systems contains machine learning and signal processing blocks that are well trained on out-of-domain data. Fig. 6.17 shows the GUI of CRSS Speech Profiler that implements the first two blocks and have intuitive visualization. It shows the audio, its spectrogram and three physical traits namely Lombard effect probability, Whisper probability, Stress probability. It also shows activation valence 2-D plot of emotion profile.

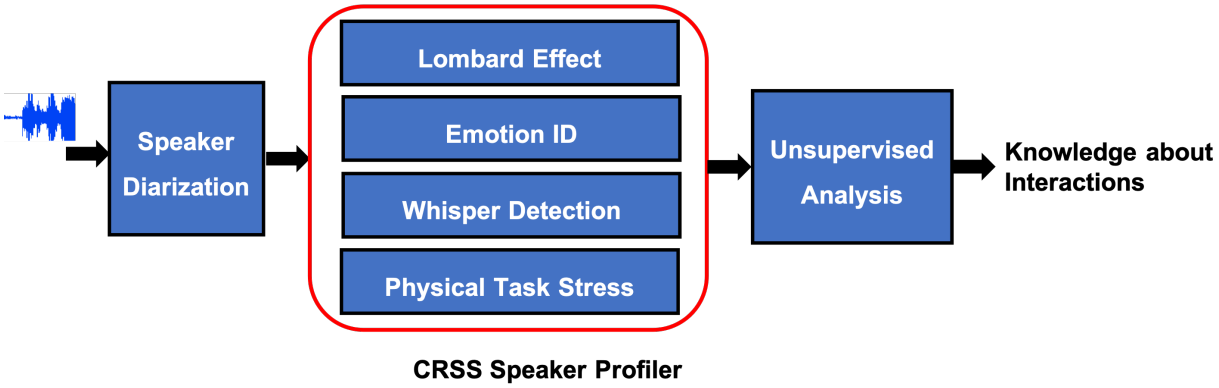


Figure 6.16. Block diagram of proposed CRSS Speech Profiler system for interaction analysis in PLTL sessions. It detects four low-level features: Emotion, Whisper, Physical Task Stress and Lombard effect. We use these features to detect engagement and communication behaviors. Speech profiler outputs the probabilities of Lombard effect, whisper and physical stress.

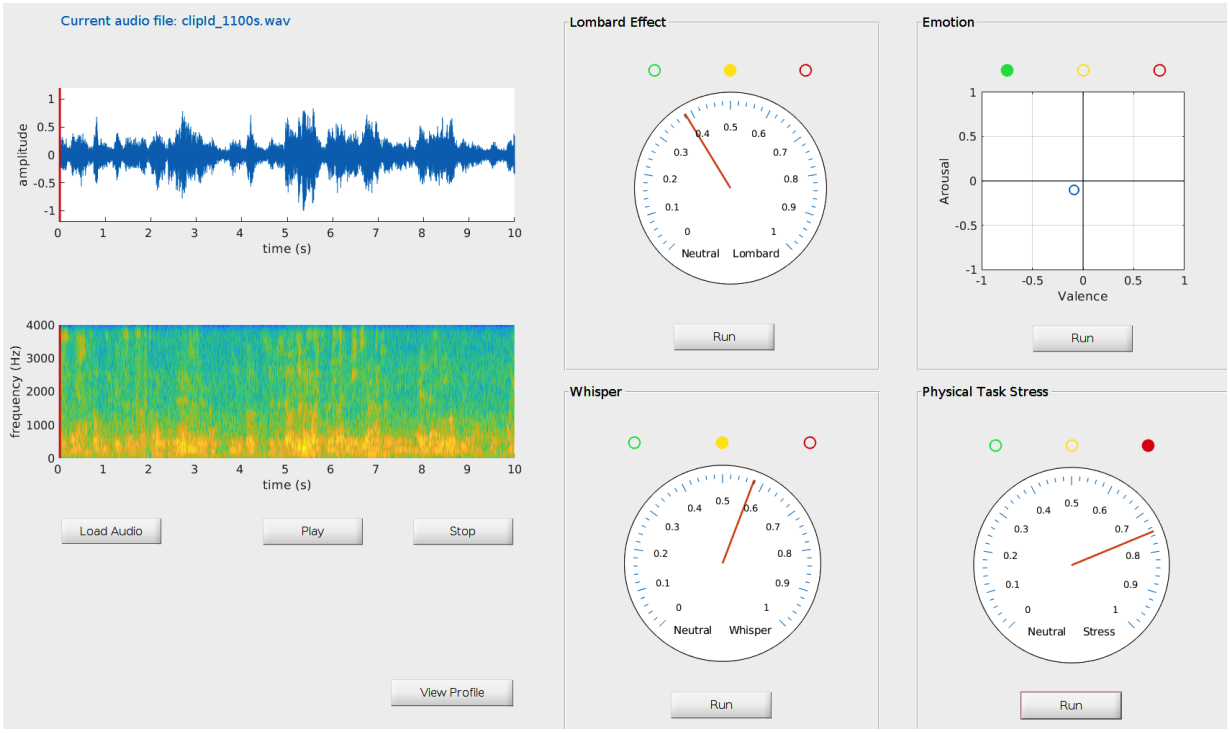


Figure 6.17. CRSS Speech Profiler graphical user interface for knowledge extraction. The three circular dots shows confidence on decisions. Red shows lower confidence and green shows high confidence while yellow shows reasonable confidence.

Emotion Recognition

Speech profiler has a emotion recognition module that is based on ladder network (Parthasarathy and Busso, 2018). This system uses unsupervised auxiliary tasks for improving recognition of emotional states. It learns the emotional attributes namely arousal, valence and dominance using a joint model. This model explores unsupervised task for regularizing the predictions. It adds unsupervised auxiliary tasks to reconstruct hidden layer representations. The auxiliary task implements denoising of hidden representations at each layer of an auto-encoder. This structure relies on ladder networks where skip connections exists between encoder and decoder layers. This framework learn powerful representations of emotional attributes. This emotion recognition system is trained using multi-task learning where it learn to predict the three emotional attributes. This model establishes state-of-the-art on MSP-Podcast corpus (Parthasarathy and Busso, 2018).

Lombard Effect

Lombard effect refers to involuntary tendency of speakers to increase their vocal effort in noisy environments for maintaining intelligible voice communication. Lombard effect can degrade the performance of speech systems (Kelly and Hansen, 2016). Proposed Lombard effect detection approach was found to be robust in presence of several noise types and their levels on UT-Scope corpus (Kelly and Hansen, 2016). CRSS speech profiler compute the probability of Lombard effect in speech segments. Here, the probability of Lombard effects shows the vocal effort of speakers (Kelly and Hansen, 2016).

Whisper Detection

Whisper detection is a measure of vocal effort found in s speech segments. CRSS speech profiler gives the probability of whisper from a speech segment. Whisper is a commonly encountered form of speech that differs significantly from modal speech. Researchers developed

i-Vector based approach for whisper detection. It was shown to perform well across different scenarios even for short speech segments (Kelly and Hansen, 2018). CRSS speech profiler outputs the probability of whisper in a speech segment. We use it as another cue for vocal effort in addition to Lombard effect.

Physical Task Stress

CRSS speech profiler detects physical task stress using a fusion of i-Vector and other speech features (Zhang et al., 2015). Physical task influence human speech production causing variabilities in speech. Such variabilities can degrade the performance of speech systems. Detecting physical task stress helps in identifying important cues from data. It leverages fusion of i-Vectors derived from MFCCs and TEO-CB-Auto-Env features for neutral/physical task stress detection (Zhang et al., 2015). MFCCs are based on linear speech production model and TEO-CB-Auto-Env features are extracted using a nonlinear operator. In this way, these features are complimentary for detecting physical task stress. This method was validated on UT-Scope physical corpus where it leads to significant accuracy gains. Furthermore, AdaBoost algorithm is used for score fusion leading to improvements in accuracy (Zhang et al., 2015).

6.14.1 Knowledge Extraction in Team Learning

This section addresses knowledge extraction for interaction analysis in team learning. We leverage low-level attributes from two categories:(i) emotion recognition- activation, valence and dominance; (ii) physical task profile- Lombard, whisper and physical task stress. These six dimensional features are mean normalized and later used in unsupervised analysis. We perform smoothed histogram based visual plotting. We perform clustering using 6D features obtained from CRSS speech profiler. Human listening experiments confirmed the two clusters to represent low and high engagement. The proposed approach for classifying engagement into

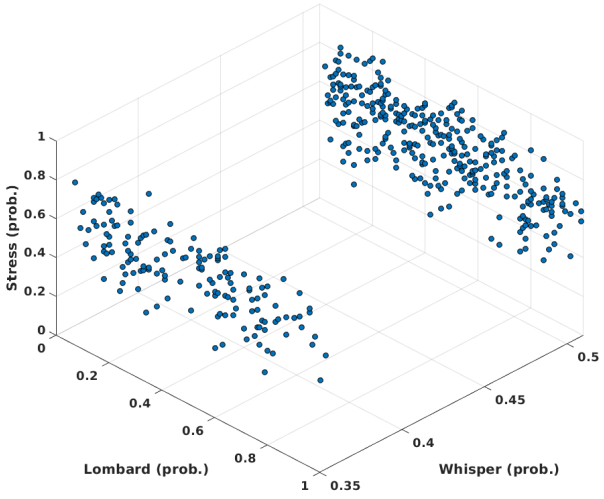


Figure 6.18. Figure showing 3D scatter plot for physical task profile (PTP). This graph is obtained from speaker segments corresponding to a PLTL session with 8 participants (80 minute). CRSS Speech profiler was used to obtain the probabilities of Lombard effect, whisper and physical task stress.

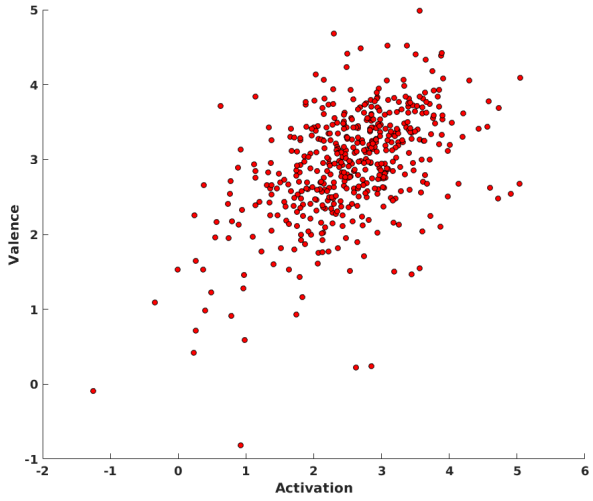


Figure 6.19. Figure showing 2D scatter plot for emotion profile with activation (x-axis) and valence (y-axis). It corresponds to 80 minute PLTL session with 8 participants. CRSS Speech profiler was used to obtain the activation and valence values for each speech segment.

low Vs high profile using thin slices of audio (10s) looks promising for knowledge extraction.

Interaction among students and their peer leader is key elements of learning in Peer Led Team Learning (PLTL). PLTL is an important learning paradigm popular in US universities for undergraduate courses. PLTL teams contains 6-8 students plus a peer leader. The team

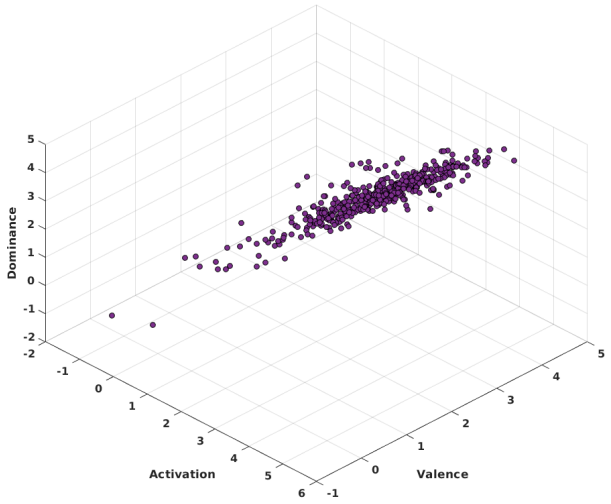


Figure 6.20. Figure showing 3D scatter plot for emotional traits profile (ETP). It corresponds to 80 minute PLTL session with 7 students and peer-lead. Speech profiler was used to obtain dominance, activation and valence values for each speech segment.

meets weekly for approx. 80 minutes sessions where students tries to address problems in collaborative ways. Peer leader's responsibilities lies in guiding the team in right direction to arrive at the solution. We assess the efficacy for proposed system on CRSS-PLTL corpus.

Fig. 6.18 shows 3D scatter plot for physical task profile (PTP). This graph is obtained from speaker segments corresponding to a PLTL session with 8 participants (80 minute). CRSS Speech profiler was used to obtain the probabilities of Lombard effect, whisper and physical task stress. Fig. 6.19 shows 2D scatter plot for emotion profile with activation (x-axis) and valence (y-axis). Fig. 6.21 illustrates the histograms for each feature in 6D attributes obtained using CRSS speech profiler. We see only probability of whisper is bi-modal. Rest features looks like a single modality.

The evaluation set for knowledge extraction using CRSS Speech Profiler consists of a PLTL session with seven students and a peer leader that accounted for 80 minute audio data. The **Silhouette Value** measures how each data point is similar to other data points in same cluster. It is used for validating the efficacy of a given clustering algorithm on a task. It is defined as $S_i = (b_i - a_i)/\max(a_i, b_i)$ where b_i is a chosen data vector and $a_i$is the average

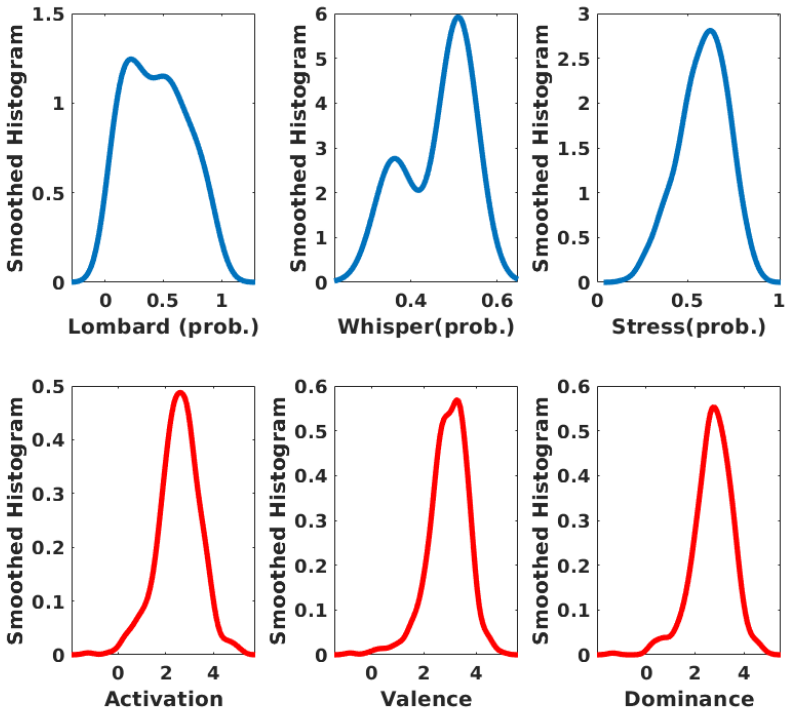


Figure 6.21. Figure illustrating histograms for each feature in 6D attributes obtained using CRSS speech profiler. We see only porbability of whisper is bi-modal. Rest features looks like a single modality.

distance from the i-th point to the other points in the same cluster as i, and b_i is the minimum average distance from the i-th point to points in a different cluster, minimized over all clusters. $S_i \in [-1, 1]$. High S_i shows that i-th data is a good match to its cluster and mis-matched from other neighboring clusters. High silhouette value shows that obtained clusters are reasonable. This criterion can adopt any distance metric and do not need ground-truth for computation. This makes it attractive for unsupervised analysis of PLTL interaction using four low-level attributes computed by CRSS speech profiler. We perform K-means clustering using 6-D features from CRSS speech profiler. Mean Silhouette values over all segments is 0.3864 for 8 clusters, 0.5973 with 2 clusters and 0.4845 for 3 clusters (See Fig. 6.22). It shows that two cluster here represents high and low engagements.

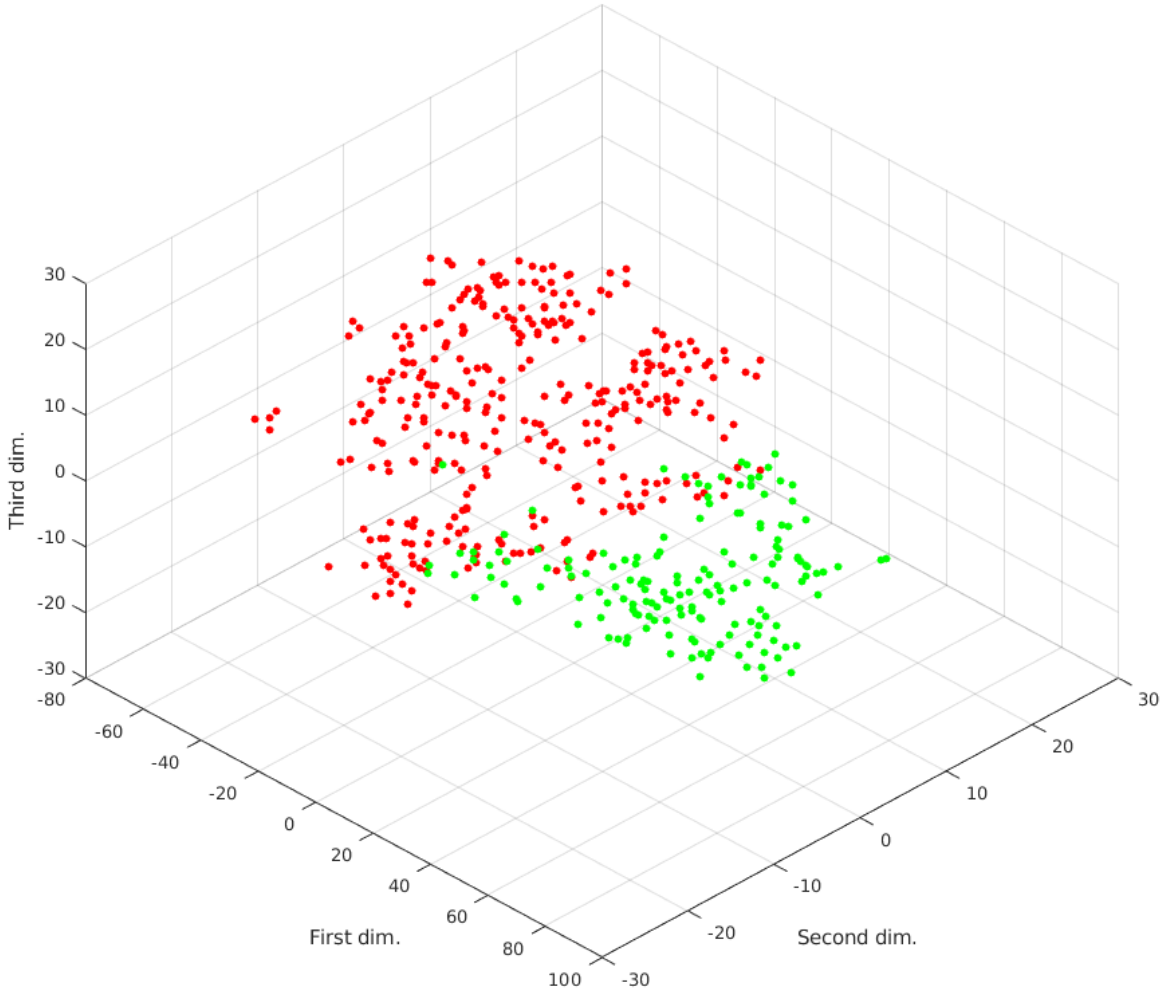


Figure 6.22. t-SNE plots with 3-D embedding of mean normalized 6-dimensional features. Color coding shows two clusters obtained using K-means. t-SNE used Euclidean distances for this plot.

6.15 Summary and Conclusions

The audio stream collected form naturalistic scenarios inherently contains rich information about person assessment, group dynamics, learning outcomes. The CRSS-PLTL corpus presents new opportunities for speech scientists and education researchers to collaborate on finding useful metrics for individual and group assessment that could be derive from audio signal. To this end, speech technology provides an opportunity for collective processing

in varied acoustic environments. Results obtained on CRSS-PLTL corpus validate the use of speech technology for knowledge extraction and interaction analysis in Peer-Led Team Learning (PLTL) groups. Even though, the development and evaluations were done on PLTL data, the proposed techniques could be used for other small-group conversations such as flipped classes, workplace meetings. The results and discussion showed the effectiveness of proposed systems for interaction analysis of PLTL sessions.

CHAPTER 7

SUMMARY AND CONCLUSIONS

Most state-of-the-art diarization techniques aim to address two-speaker structured diarization which is significantly simpler as compared to free-from small-group with 8-10 individuals scenario. Furthermore, most state-of-the-art systems are developed for telephone speech which is both clean and structured conversations simpler for diarization than PLTL-type naturalistic audio. This dissertation provided solutions for robust speaker diarization of small-group conversations (4-10 speakers) in naturalistic audio streams. As noted, naturalistic audio has uncontrolled noise, reverberation, overlap and other acoustic events that degrade performance of state-of-the-art systems. In this dissertation, we studied three important blocks in diarization pipeline: (i) speech activity detection (SAD), (ii) speaker modeling for speaker recognition and diarization, and (iii) speaker clustering. Specifically, SAD, speaker recognition and modeling with SincNet convolutional neural network (CNN) and model-based speaker clustering were investigated. A probe study also explored knowledge extraction and interaction analysis in Peer-Led Team Learning (PLTL) sessions. To this end, we proposed several techniques for analyzing the PLTL conversations that can potentially help PLTL researchers.

This chapter covers the specific dissertation contributions including highlights of algorithmic advancements and their significance for naturalistic scenarios. We close the chapter by providing pointers for future work.

7.1 Dissertation Contributions

As part of the advancements, naturalistic audio corpora were established including: CRSS-PLTL (Dubey et al., 2016), CRSS-PLTL-II (Dubey et al., 2017), and CRSS-LDNN (Hansen et al., 2018). CRSS-PLTL and CRSS-PLTL-II contains audio recordings of five teams attending undergraduate Chemistry and Calculus courses at The University of Texas at Dallas. These

corpora were collected for 11 weeks in two different semesters. CRSS-LDNN contains long-duration noise recordings from naturalistic scenarios where more than one noise source was present. The secondary probe efforts considered knowledge extraction for interaction analysis of Peer-Led Team Learning (PLTL) groups. **Contribution #1: Frequency-Dependent Kernel (FI**

We proposed FDK features as a novel way to decompose speech signals where distinct frequency-dependent kernels are used for analyzing different frequency bins. We employed frequency-dependent Gaussian kernels where the width of the kernel were inversely proportional to frequency bin. In this way, we obtain narrow kernels for higher frequencies and wider ones for lower frequencies. FDK features aim to provide a generalized decomposition of signal energies across different time-frequency locations. We derived eight statistical descriptors from the logarithm of the absolute value of the FDK vector corresponding to each frame. These statistical descriptors were also mean and variance normalized. The resulting normalized features were later processed with principal component analysis (PCA), where the first component chosen as the final FDK-SAD feature. This FDK-SAD feature was incorporated with three proposed decisions backends for achieving unsupervised/semi-supervised SAD. Results from Sec. 3.9.4 demonstrated that FDK-SAD out-performed SohnSAD by +92.87% for CRSS-PLTL data. Further SAD experiments are reported in Tables 3.2, 3.3, 3.4 and Sec. 3.10.5.

Contribution #2: SAD Decision Backends (i) VMGMM; (ii) DipSAD, and (iii) D-SAD

We proposed three decision backends for SAD that include: (i) Variable Model Size Gaussian Mixture Model (VMGMM); (ii) Hartigan Dip test for robust feature clustering (DipSAD), and (iii) Density-SAD (D-SAD). VMGMM uses Akaike information criterion (AIC) for estimating the model order of a GMM used to model the SAD features. GMM means were combined in

a convex manner to form a general SAD threshold. The weights for combining the GMM means are user defined and can be optimized based on development data. Thus, VMGMM is a semi-supervised approach. In an attempt to design a fully unsupervised SAD; we leveraged Hartigan dip test recursively for segmenting (clustering) SAD features into speech and non-speech clusters, resulting in the second approach called DipSAD. Next, we proposed the third method called Density-SAD (D-SAD) as a computationally simple and fully unsupervised SAD. D-SAD fits a straight line by joining the first and last data points in a cumulative distribution curve (CDC) for the SAD features. The point of intersection between straight line and CDC curve defines the operating decision threshold. We combined three alternate decision backends with FDK-SAD feature to obtain three unsupervised/semi-supervised SAD systems. We performed comparative studies (see Table 3.6) to highlight the competitive trade-offs of the proposed SAD techniques over prior state-of-the-art approaches.

For evaluation, we used a DCF equal weight (0.5) for false-alarms and miss rate, since both components are equally important for diarization. This fact is illustrated by DER formula (see Eq. 2.3). We obtain following DCF (%) for CRSS-PLTL data: Combo-VMGMM (1.97%) where Combo-SAD features were combined with VMGMM backend, FDK-VMGMM (2.01%) where FDK-SAD features were combined with VMGMM backend, Combo-DipSAD (2.84%) where Combo-SAD features were combined with DipSAD backend, FDK-DipSAD (7.23%) where FDK-SAD features were combined with DipSAD backend, FDK-DSAD (15.29%) where FDK-SAD features were combined with DSAD backend, Combo-DSAD (17.68%) where Combo-SAD features were combined with D-SAD backend. We see that D-SAD gives worse performance as compared to VMGMM and DipSAD. However, it is still superior than baseline methods namely SohnSAD (28.20%), rSAD (49.57%), SSGMM (28.95%), TO-Combo-SAD (29.16%) and USC-DNN-SAD (23.19%) which is neural networks based supervised SAD approach. More details on these results can be found in Sec. 3.10.4.

We performed experiments involving standalone SAD and text-dependent speaker verification for redDots corpus. The proposed features were found to perform consistently well

on both tasks leading to significant relatives gains in EER (%). We compared baseline SAD approaches with proposed ones for text-dependent speaker verification on redDots data. We obtained the following EER: No SAD (7.90%) where silences were not discarded, SohnSAD (6.32%), Combo-VMGMM (6.97%) where Combo-SAD features were combined with VMGMM decision backend, Combo-DipSAD (10.12%) where Combo-SAD features were combined with DipSAD backend, FDK-VMGMM (6.29%) where FDK features were combined with VMGMM backend, FDK-DipSAD (9.48%) where FDK-SAD features were combined with DipSAD backend, rSAD (6.58%), SSGMM (7.37%). For more details on these experiments, please refer Sec. 3.10.5.

Contribution #3: Speaker Modeling (i) SincNet,(ii) AM-SincNet, (iii) CL-SincNet, and (iv) AM-CL-SincNet

We proposed raw waveform modeling with SincNet convolutional neural network for speaker modeling in diarization pipeline.This architecture is trained for frame-level (10ms) speaker identification. We proposed novel AM-SincNet, CL-SincNet and AM-CL-SincNet by incorporating discriminative loss functions: (i) additive margin (AM)-Softmax; and (ii) Center Loss (CL). These architectures improve SincNet based speaker modeling. We leveraged recently developed discriminative loss functions such as additive margin (AM)-Softmax, and Center Loss (CL) to advance the standard SincNet architecture to AM-SincNet, CL-SincNet, and AM-CL-SincNet. We investigated supervised transfer learning (STL) for improving the generalization ability of SincNet to multiple data sets. STL reduces the training time of proposed SincNet architectures. The STL approach first trains a SincNet model for speaker recognition on TIMIT data. Later, we adopted this model and discarded its output layer. It is followed by adding two new layers (fully connected hidden layer and output layer for Librispeech training data) to already trained TIMIT SincNet model. This network is now re-trained for learning the parameters of newly added layers, i.e., last two layers while parameters of

other pre-trained layers is kept fixed. This strategy is based on the fact that earlier layers of SincNet neural network tries to learn robust features for speaker representation while later layers learn domain-specific speaker discrimination information. This STL approach leads to 100 times improvements in convergence speech, i.e., it reduces the training time by order of 100. STL SincNet architecture leads to better results on both in-domain speaker recognition (Librispeech test data) and speaker diarization on CRSS-PLTL and AMI corpora. We trained SincNet architectures using out-of-domain data such as TIMIT and Librispeech corpora. Trained SincNet was adopted for unsupervised transfer learning (UTL) where we extract frame-level speaker embeddings from in-domain CRSS-PLTL and AMI data. We also performed experiments with supervised transfer learning (TL) for data efficient training of SincNet. TL approach trains the SincNet model first using TIMIT, next we discarded the output layer and added two new layers for training on Librispeech corpus. We optimized the hyper-parameters and discussed its importance in achieving robust diarization performance.

SincNet and its variants extract speaker embeddings from short speech segments of size 100-200ms with 10ms skip. This approach eliminates the need of speaker change detection. SincNet embeddings are found to be superior than i-Vectors as i-Vector do not perform well for short utterances. Proposed novel SincNet architectures converge faster that standard SincNet. The neural speaker modeling using SincNet architecture was found to perform significantly better than i-Vector baseline. The proposed SincNet architecture are robust even with limited amount of training data (15 second/speaker). The advancements in discriminative loss function and supervised transfer learning (STL) helps in improving the convergence speed and hence the total training time. By using proposed approach, we out-performed state-of-the-art speaker recognition on Librispeech corpus. This dissertation contributed novel SincNet architectures for speaker recognition and diarization. We studied the effect of SAD on SincNet based speaker diarization and found that SAD is very important for robustness of diarization performance on naturalistic audio such as CRSS-PLTL. We studied the effect of

margin parameter in AM-SincNet for speaker diarization. There was no consistent trend in DER for change in margin parameter, m that lies in range [0,1]. We found that AM-SincNet with $m = 0.9$ gave best performance in terms of DER where the AM-SincNet was trained on TIMIT data. We also studied the effect of CL parameters on DER and could not find a regular trend. We noticed that adding center loss (CL) to AM-SincNet, i.e., AM-CL-SincNet makes the DER robust to changes in CL parameters. We studied different ways to extract parameters from trained SincNet. We found that average pooling leads to better diarization performance than max pooling where pooling was done over all embeddings from a segment to convert frame-level embeddings to segment-level embeddings. We found that F2 embeddings (output of last convolutional layer) was better than F1 embeddings (activations of output layer) and F3 embeddings (output of Sinc Layer). Standard SincNet lead to DER of 12.81% as compared to i-Vector DER of 15.26. When, we used AM-SincNet with $m = 0.90$, DER reduces to 8% that shows effectiveness of discriminative loss functions. Similarly, we obtained significant DER reductions for 12-meetings set of AMI corpus. More results and discussions can be found in Sec. 4.8.2.

In addition to SincNet based speaker modeling, we reviewed our initial work that includes unsupervised denoising autoencoder (DAE) for meeting-specific speaker embedding extractor and HMM for joint segmentation and speaker clustering.

Contribution #4: Speaker Clustering (i) movMF, (ii) NFCM, and (iii) TIC

We proposed three model-based approaches for speaker clustering. The proposed clustering methods rely on established theoretical foundations and structural constraints present in length-normalized speaker embeddings. First method leverages mixture of von Mises-Fisher distributions (movMF) for clustering length-normalized speaker embeddings. In this case, each component in movMF mixture model represents a speaker. Standard expectation maximization (EM) was used for iterative speaker clustering that alternates between cluster

assignment (in E-step) and re-estimation of movMF model parameters (in M-step). Second approach is based on normalized Fuzzy C-means (NFCM) clustering which is suitable for length-normalized speaker embedding. It is motivated from recent developments on Fuzzy C-means based soft-clustering of length-normalized data. Soft speaker clustering provides possibility of flexible decision making in diarization pipeline. Standard NFCM assigns the speaker embeddings to a cluster that maximizes the likelihood of data given the model. Our third approach is more sophisticated and computationally expensive as compared to movMF and NFCM. Toeplitz Inverse Covariance (TIC) speaker clustering tries to learn a Markov Random Field (MRF) correlation network for each speaker. It models each speaker using a Toeplitz Inverse Covariance matrix.We rely on dynamic programming (DP) for cluster assignment. The clustering problem is essentially a Toeplitz graphical lasso optimization problem. We conducted several experiments to benchmark the proposed clustering approaches with respect to cosine K-means baseline. Speaker embeddings (F2-avg) from AM-SincNet with margin parameter (m=0.95) leads to following DER (%): for Cosine K-means (15.07%), movMF (14.58 %), NFCM (10.58 %), and TIC (14.04 %). When we used i-Vector feature of dimensions 75, these methods lead to following DER (%): for Cosine K-means (17.15%), movMF (12.96%), NFCM (12.5%) and TIC (8.88%). We can see proposed methods leads to significant relative DER reductions of upto 48.22%. Similarly, the proposed methods significantly out-performed cosine K-means baseline for 12 meetings subset of AMI corpus where DER for many meetings was in order of 1-2%. These experiments validated the efficacy of proposed methods for robust speaker clustering in naturalistic scenarios.

Contribution #5: Knowledge Extraction and Interaction Analysis in PLTL Sessions

This was the secondary focus of this dissertation where we started from diarization output and performed analysis to extract knowledge about conversational dynamics and behavioral

metrics related to PLTL sessions. We proposed novel metrics to achieve this task. Specifically, we proposed (i) unsupervised dominance score, (ii) question inflection detection, (iii) emphasis detection, (iv) speech rate (word), and (v) CRSS Speaker Profiler based analysis. CRSS Speaker profiler detects four low-level features namely (1) Emotion, (2) Whisper, (3) Physical Task Stress and (4) Lombard effect. These features were later utilized in unsupervised analysis for inferring engagement in communication behavior of students participating in PLTL sessions. The results showed validity of proposed approach for engagement detection and interaction analysis in PLTL sessions. More details on these experiments and discussions can be found in Sec. 6.14.1.

7.2 Future Work

This dissertation contributed several advancements for robust speaker diarization. However, it is just the beginning of second generation diarization systems based on Machine Learning (ML) models. There are some aspects of proposed research that can be extended into related tasks as well.

1. **Benchmarking frequency-dependent kernel (FDK) for other speech tasks:** The frequency-dependent kernel (FDK) features are derived from a decomposition matrix that can be utilized as a substitute for spectrogram or Mel-spectrogram for automatic speech recognition (ASR), speaker identification, language and dialect identification. Instead of deriving eight features from FDK spectrum $D(\tau, f, \theta)$, we can just take $\mathbf{E} = 20 \log 10(|\mathbf{D}|)$ from Eq. 3.1 as described in Sec. 3.5. The FDK log spectrum $\mathbf{E}$ can replace Mel-spectrum in speech tasks. Mel-scale was developed to mimic human auditory perception, however machine learning models can have different ways to perceive the input feature. There is no sufficient evidence that Mel-spectrum is best for all deep learning models for speech and audio recognition tasks.

2. **Parametric D-SAD**:

   Cumulative Distribution based SAD (D-SAD) is parameter free approach. However, we can introduce a parameter that can help in tuning D-SAD backend on different domains or tasks. We can name the D-SAD with an additional intercept parameter c as parametric D-SAD (pD-SAD). As we have described in Sec. 3.8, D-SAD fits a straight line between first and last point in cumulative distribution curve (CDC). Lets say, $feats_{min}$ and $feats_{max}$ are minimum and maximum value of features as extracted from CDC. We compute the slope of straight line connecting the points [$feats_{min}$, 0] and [$feats_{max}$, 1]. The slope, m is given as

$$m = \frac{1}{(feats_{max} - feats_{min})} \tag{7.1}$$

Now, we define the straight line for parametric D-SAD as:

$$y = m \cdot x + c \tag{7.2}$$

where x are feature values that lies in range [$feats_{min}$, $feats_{max}$]. The parameter c is introduced to make this model flexible. We can derive a new straight line for each value of c. Each of such lines are parallel to each other. The point of intersection of these line with CDC makes a new SAD threshold if such intersection point exists. Thus, we can get different SAD threshold for each unique value of c. In this way, we can compute DCF for different SAD threshold and choose the parameter c that can minimize the DCF. We see that this approach helps in optimizing DCF for a given application where some development data is available. It is possible that D-SAD and pD-SAD are good solutions for any binary classification with one-dimensional features.

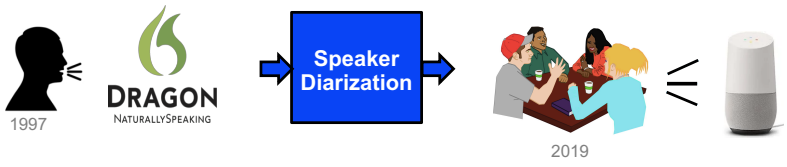
DEEP NEURAL NETWORKS AND MODEL-BASED APPROACHES FOR ROBUST SPEAKER DIARIZATION IN NATURALISTIC AUDIO STREAMS

Harishchandra (Hari) Dubey

Robust Speech Technologies Lab (RSTL)
Center for Robust Speech Systems (CRSS)
Erik Jonsson School of Engineering & Computer Science
Department of Electrical Engineering
University of Texas at Dallas
Richardson, Texas, 75083-0688, U.S.A.

Ph.D. Dissertation Defense
May 1, 2019 ECSN 2.704

Motivation



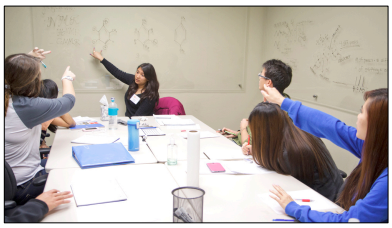
- Diarization answers "who spoke and when"
- Primary speaker Vs. secondary speaker
- Front-end for multi-speaker speech services such as ASR, NLP

Small-group Conversations

Augmented Multi-modal Interaction

(a) AMI meeting corpus

UTDallas Student Success Center: PLTL



(b) PLTL

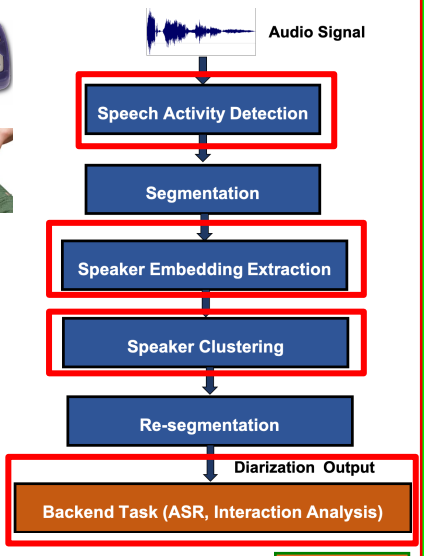
- **Goal: advancing Diarization for 4-10 speakers**
  - Meeting recordings, Peer-Led Team Learning (PLTL) groups
- **Knowledge extraction for interaction analysis**
  - speech - > Inference, Analysis etc.

Diarization: Investigated Tasks

LENA

Wearable Pouch

- **Research Goals**
  - Corpora: ~8 speakers
  - Propose Novel Algorithms
- **Studied Sub-tasks**
  - Speech Activity Detection (SAD)
  - Neural Speaker Modelling
  - Speaker Clustering
  - Interaction Analysis



Dissertation Contributions

- **CONTRIBUTION #1: Speech Activity Detection (SAD)**
  - Feature: FDK-SAD
  - Decision Backends: VMGMM, DipSAD, D-SAD
- **CONTRIBUTION #2: SincNet based Neural Speaker Modelling**
  - Unsupervised Transfer Learning (TL), Supervised TL
  - SincNet, AM-SincNet, CL-SincNet, AM-CL-SincNet
- **CONTRIBUTION #3: Robust Speaker Clustering**
  - movMF: Mixture of von Mises-Fisher distributions
  - NFCM: Normalized Fuzzy C-means clustering
  - TIC: Toeplitz Inverse Covariance-based speaker clustering
- **CONTRIBUTION #4: Interaction Analysis**
  - Speaker Profiler, Engagement, Dominance Score

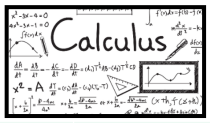
Outline

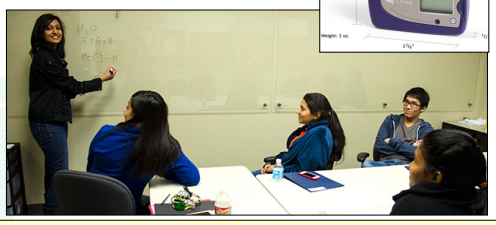
- **1.0 Speech Activity Detection (SAD)**
  - FDK-SAD: **F**requency-**D**ependent **K**ernel Features
  - VMGMM: **V**ariable **M**odel-size **GMM** Decision Backend
  - Dip-SAD: Hartigan **Dip** Test-based Decision Backend
  - D-SAD: Cumulative **D**ensity based linear curve Decision Backend
- **2.0 Raw Waveform SincNet for Speaker Modelling**
  - Speaker ID and Diarization with Transfer Learning (TL) and SincNet
  - Discriminative loss functions: Additive Margin (AM)-Softmax and Center Loss (CL)
- **3.0 Model-based Speaker Clustering**
  - movMF: Mixture of von Mises-Fisher distributions
  - NFCM: Normalized Fuzzy C-means clustering
  - TIC: Toeplitz Inverse Covariance-based speaker clustering
- **4.0 Interaction Analysis**: Speaker Profiler, Dominance Score

CRSS-PLTL Corpus

- 7-9 members per PLTL team; Each member wears LENA audio recorder. 11 sessions per team (1/week); 5 teams.
- Each session: 80 min. Team engaged in problem solving.
- Post Eval: STEM-Chemistry & Calculus2: Each member surveyed to measure engagement & motivation after each session

(Corpus Update: Chemistry: Spring'16; Calculus: Spring'17)





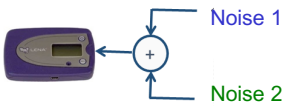
Total Days	11
Total groups	5
Total Recorded Sessions	55
Audio Data Duration (Chem)	330-340 Hrs
Audio Data Duration (Calc2)	300-400 Hrs

[1] **H. Dubey**, L. Kaushik, A. Sangwan and J. H. L. Hansen, "A Speaker Diarization System for Studying Peer-Led Team Learning Groups", *ISCA INTERSPEECH, 2016*, San Francisco, USA, pp. 2180–2184, 2016.

CRSS-Long Duration Naturalistic Noise (LDNN) Corpus

- Rich variety of multi-layer Noise, e.g., bus-engine + babble+ impulsive
- Approx. 19 hours duration, 16 kHz sampling rate, 16-bit resolution
- 3 categories: Non-stationary, Impulsive, Harmonic
- Only Noise is recorded

Water Tap



UTD construction site

Babble, Large-Crowd

UTD Bus

Household Equipment

[1] J. H. L. Hansen, **H. Dubey**, and A. Sangwan, "CRSS-LDNN: Long-duration naturalistic noise corpus containing multi-layer noise recordings for robust speech processing", *The Journal of the Acoustical Society of America, 2018*, 144(3), pp.1797-1797.

Exploratory Data Analysis: PLTL

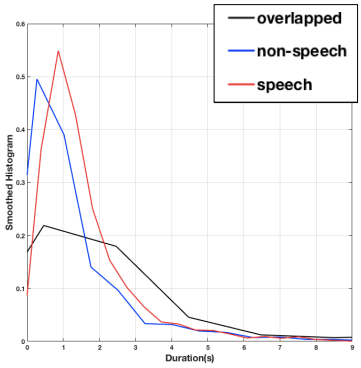


Fig: (a) Distribution of segment duration. Contains Speech (41.72%); Overlapped (28.71%; Non-speech (29.57%)

(dB)

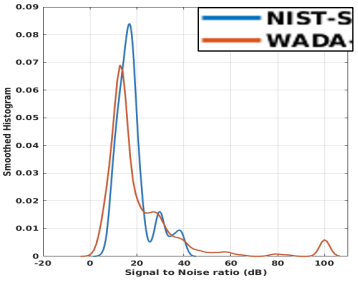


Fig: (b) High SNR shows low-mod noise

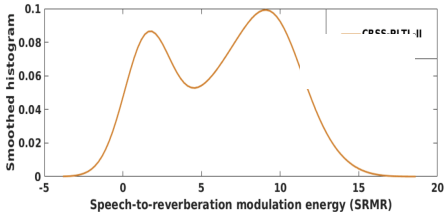


Fig: (c) Low SRMR shows reverberation

[1] Falk, Tiago H., Chenxi Zheng, and Wai-Yip Chan, " A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech ", *IEEE Trans. on Audio, Speech, and Language Processing (2010):* 18.7 1766-1774

Error Metrics: DER & SAD-DCF

- SAD Detection Cost Function (DCF), w= 0.5 for Diarization as miss rare and false alarms are equally important (see DER eqn.)

$$DCF = w * P_{\{miss\}} + (1 - w) * P_{\{fa\}}$$

Weight — Miss rate — False-alarm rate

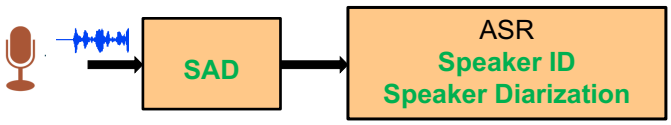
- Diarization error rate (DER) given as

False-alarm (SAD) — Miss (SAD) — Speaker error

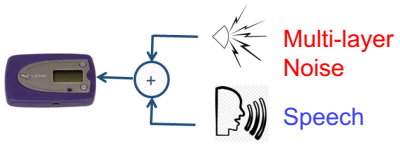
$$DER = \frac{t_{\{miss\}} + t_{\{fa\}} + t_{\{err\}}}{t_{\{total\}}}$$

Total-time from reference segments

SAD for Naturalistic Scenarios



- Traditional: SAD algorithms tested on 1-D noise
- Naturalistic: Multi-layer noise



- **Target applications**
  - Naturalistic audio
  - Low-resource speech systems

- Performance of SAD directly impacts follow-on speech tasks
- Unsupervised SAD offers flexibility for diverse changing environments

State-of-the-art SAD: Methods

- **Phonetic-SAD [1]:** A two-stage approach, In first stage a bottleneck DNN is trained to predict senone posteriors. In second stage, activations at bottleneck layer used to train a second DNN for SAD. **(Supervised Methods)**
- **USC-DNN-SAD [2]:** 4-D features namely (i) spectral shape, (ii) spectro-temporal modulations, (iii) periodicity, and (iv) the long-term spectral variability are used for training a feedforward DNN for SAD.
- **SohnSAD [3]:** Decision rule derived from the generalized likelihood ratio test that considered the geometric mean of likelihood ratios over all frequency bins **(Unsupervised Methods)**
- **rSAD [4]:** High-pass filtering followed by selection of high-energy frames based on aposteriori SNR weighted energy-difference. The high-energy frames without a pitch value in reasonable range were taken as non-speech.
- **SSGMM [5]:** Semi-supervised GMM, trains two GMMs for speech & non-speech. Higher energy frames used to train speech GMM and lower energy frames for non-speech.
- **Combo-SAD [6]:** Handcrafted 5-D features namely harmonicity, clarity, prediction gain, periodicity and spectral flux. Processed with PCA 1-dim and considered in 2-GMM for decision.

[1] L. Ferrer, M. Graciarena, and V. Mitra, "A phonetically aware system for speech activity detection," *IEEE ICASSP, 2016*.
[2] M. Van Segbroeck, A. Tsiartas, S. Narayanan, "A robust frontend for VAD: exploiting contextual, discriminative and spectral cues of human voice", *ISCA INTERSPEECH, (2013)*, pp. 704-708.
[3] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters, (1999)*, vol. 6, no. 1, pp. 1–3.
[4] Z.-H. Tan and B. Lindberg, "Low-complexity variable frame rate analysis for speech recognition and voice activity detection." *IEEE Journal of Selected Topics in Signal Processing, (2010)*, vol. 4, no. 5, pp. 798-807.
[5] A. Sholokhov, M. Sahidullah, and T. Kinnunen,"Semi-supervised speech activity detection with an application to automatic speaker verification," *Computer Speech & Language, (2018)*, vol. 47, pp. 132–156.
[6] S. O. Sadjadi and J. H. L. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *IEEE Signal Processing Letters, (2013)*, vol. 20, no. 3, pp. 197–200.
[7] A. Ziaei, L. Kaushik, A. Sangwan, J. H. L. Hansen, D.W. Oard, "Speech activity detection for NASA Apollo space missions: Challenges and solutions ", *ISCA INTERSPEECH, (2014)*.

Frequency Dependent Kernel (FDK-SAD)

- **Motivation:**
  - Under adverse acoustics conditions (e.g., multi-layer noise), there could be possibly many processes in audio generation
  - Leverage frequency-dependent Gaussian kernels (FDGKs) for spectral decomposition
- **Advantages:**
  - Better than other kernels due to its compact time-frequency spectrum. For Gaussian kernels, the product of uncertainty in time and frequency domains is minimum.
  - Variance in FDGK is inversely proportional to frequency i. e., higher width for lower frequencies
- FDK provides generalized spectral decomposition that quantifies multiple noise-sources and speech contained in an audio signal

Frequency Dependent Kernel (FDK-SAD)

- FDK decomposition is defined below

Audio Signal

$$D(\tau, f, \theta) = \int s(t) w(\tau - t, f, \theta) \exp(-j2\pi f t) dt,$$

- To ensure consistency, windows should satisfy:

$$\int w(\tau - t, f, \theta) d\tau = 1$$

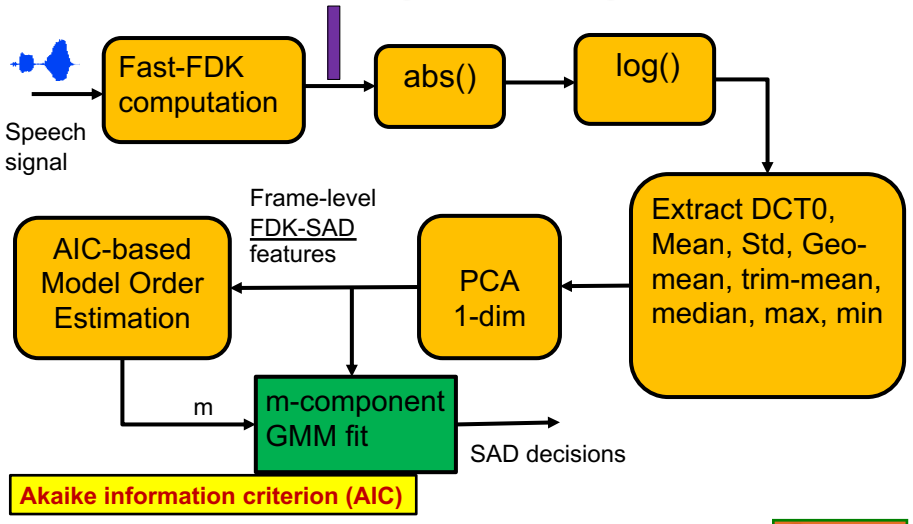
- Gaussian kernels with frequency-dependent variance

$$w_{Gauss}(\tau - t, f, \sigma) = \frac{|f|}{\sqrt{2\pi}\sigma} \exp(-\frac{f^2(\tau - t)^2}{2\sigma^2})$$

- $D(\tau, f, \theta)$ is generalized spectral decomposition
- Captures speech non-speech statistics in an audio

[1] **H. Dubey**, A. Sangwan, and J. H. L. Hansen, "Leveraging Frequency-Dependent Kernel and DIP-Based Clustering for Robust Speech Activity Detection in Naturalistic Audio Streams", *IEEE/ACM Trans. on Audio, Speech, and Language Processing, (2018):* 26.11, 2056-2071.".

Frequency Dependent Kernel (FDK-SAD)



Variable Model-size GMM (VMGMM)

- Under adverse acoustic conditions such as multi-layer noise, SAD features (FDK/Combo) do not remain bimodal
- Akaike information criterion (AIC) for estimating the model-order of GMM

$$AIC = -2\log \mathcal{L}(\hat{\theta}|y) + 2k$$

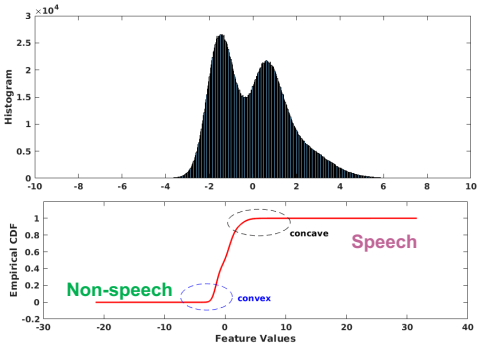
Likelihood function

- k is the number of estimated parameters and y is data, theta stands for ML estimates of parameters
- BEST model is one with minimum AIC value

$$\tau_{mGMM} = w_1\mu_1 + w_2\mu_2 + \cdots + w_m\mu_m$$
$$w_1 + w_2 + \cdots + w_m = 1$$

[1] **H. Dubey**, A. Sangwan, and J. H. L. Hansen, "Leveraging Frequency-Dependent Kernel and DIP-Based Clustering for Robust Speech Activity Detection in Naturalistic Audio Streams", *IEEE/ACM Trans. on Audio, Speech, and Language Processing, (2018):* 26.11, 2056-2071.".

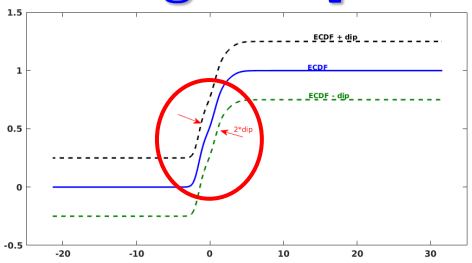
Cumulative Distribution



- **Top Figure:**
  - Histogram of Combo-SAD features from an utterance (audio recording)
- **Bottom Figure:**
  - Cumulative distribution (CDF) of features follows the convex-then-concave nature when the *mode* changes from Non-speech to Speech

[1] **H. Dubey**, A. Sangwan, and J. H. L. Hansen, "Leveraging Frequency-Dependent Kernel and DIP-Based Clustering for Robust Speech Activity Detection in Naturalistic Audio Streams", *IEEE/ACM Trans. on Audio, Speech, and Language Processing, (2018):* 26.11, 2056-2071.

Hartigan Dip Test



- A distribution is unimodal if its CDF takes a convex form up to its mode/modal interval and a concave form after it.
- Dip statistic, **DIP ∈ (0, 1/4],** is defined as the minimum achievable vertical offset for two copies of ECDF (**one above**, **one below**, shown in dashed lines) such that piece-wise linear-fit (of ECDF) does not violate its unimodal rule
  - Farther distribution move away from unimodality, larger the corresponding dip

[1] P. M. Hartigan, "Algorithm AS 217: Computation of the dip statistic to test for unimodality" *Journal of the Royal Statistical Society, Series C (Applied Statistics) 34.3 (1985):* 320-325.

DipSAD

Algorithm 1 *computeDip*

Input: speech features were sorted in ascending order i.e., **o**=$[o_1, o_2, ..., o_N]$ where $o_1 \leq o_2 \leq ... \leq o_N$.
Output: primary modal interval $[o_L, o_U]$, DIP and p-value, p.

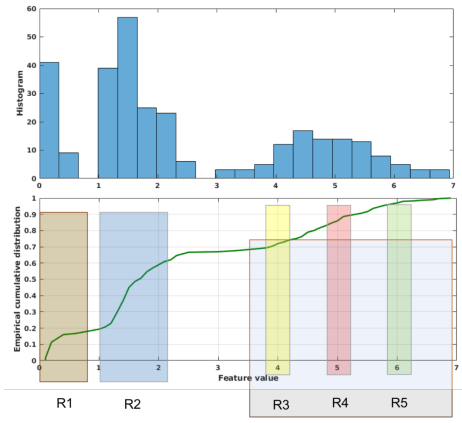
Step 1: Initialize, lower point $o_L = o_1$, upper point $o_U = o_N$ and D = 0.
Step 2: Compute greatest convex minorant G and least concave majorant H of empirical distribution F of features in interval $[o_L, o_U]$ [2]. Let the points of contact with F are respectively, $g_1, g_2, .., g_k$ (for G) and $h_1, h_2, .., h_m$ (for H).
Step 3: Let d= max $|G(g_i)$ - $H(g_i)| >$ max $|G(h_j)$ - $H(h_j)|$ and the maximum occurs at $h_j \leq g_i \leq h_{j+1}$. Then, define $o_L^0 = g_i$, $o_U^0 = h_{j+1}$.
Step 4: Let d= max $|G(h_j)$ - $H(h_j)| \geq$ max $|G(g_i)$ - $H(g_i)|$ and the maximum occurs at $g_i \leq h_j \leq g_{i+1}$. Then, define $o_L^0 = g_i$, $o_U^0 = h_j$.
Step 5: If $d \leq D$, Stop and set DIP= $\frac{D}{2}$.
Step 6: If $d > D$, set D= max {

$$\sup_{o_L \leq o \leq o_L^0} |G(o) - F(o)|, \quad \sup_{o_U \leq o \leq o_U^0} |H(o) - F(o)|$$

}, where sup is the supremum (supremum is the smallest number that is greater than or equal to every number in the set).
Step 7: Set $o_L = o_L^0$, $o_U = o_U^0$. Go to Step 2.

[1] **H. Dubey**, A. Sangwan, and J. H. L. Hansen, "Leveraging Frequency-Dependent Kernel and DIP-Based Clustering for Robust Speech Activity Detection in Naturalistic Audio Streams", *IEEE/ACM Trans. on Audio, Speech, and Language Processing, (2018):* 26.11, 2056-2071.".

DipSAD : Illustration



- **Examples with 5 classes:**
  - First iteration gives: [R3,R5] and [R1,R2]
  - Second iteration within [R3,R5] gives R3,R4 and R5
  - Iterating in [R1,R2] gives R1 and R2

- 1. We included nearest region in left-search and right-search
- 2. Dip-SAD on a sorted feature-vector requires **O(N)** operations in the worst case; where N is number of frames

DipSAD

Algorithm 2 *Dip-SAD*

Input: frame-level speech features from an utterance
Output: speech non-speech labels for each frame

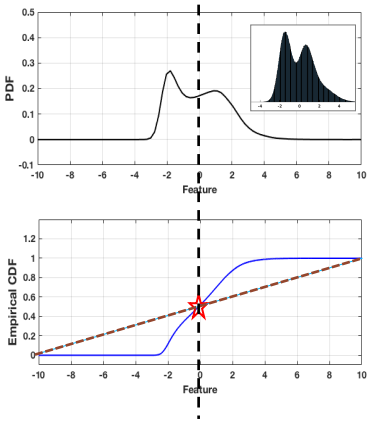
Step 1: Sort the features in ascending order and let $\mathbf{o}=[o_1, o_2, ...,o_N]$ be the ordered vector, where $o_1 \leq o_2 \leq ... \leq o_N$. The significance level, α is set to 0.05 for all experiments reported in this paper.
Step 2: $\{o_L, o_U, p\} \leftarrow computeDip(\mathbf{o})$
Step 3: If $p > \alpha$, then the detected primary modal interval is $[o_L, o_U]$. Else, $[o_1, o_N]$ is primary modal interval.
Step 4: Recurse into the modal interval to find the list I_{mid} of the modal intervals within detected primary mode.
Step 5: Now, we check to the right and left of the primary modal interval recursively and extract additional modes if found.
Step 6: $\{u\} \leftarrow \min_{o_U \in I_{mid}} (o_U)$, $\{l\} \leftarrow \max_{o_L \in I_{mid}} (o_L)$.
Step 7: $p_l \leftarrow computeDip(\ \forall o_j : o_j \leq u)$, $p_u \leftarrow computeDip(\ \forall o_j : o_j \geq l)$.
Step 8: $I_l \leftarrow$ If $p_l \leq \alpha$, then $\forall o_j : o_j < o_l$ forms a multi-mode segment. We recurse into this interval and return all found modal intervals. Else return ϕ i.e., an empty set.
Step 9: $I_r \leftarrow$ If $p_u \leq \alpha$, then $\forall o_j : o_j > o_u$ forms a multi-mode segment. We recurse into this interval and return all found modal intervals. Else return ϕ i.e., an empty set.
Step 10: The final set of all modal interval is $I_l \bigcup I_{mid} \bigcup I_r$.
Step 11: As we knew that combo-SAD features have high positive value for speech and low value for different noises, the cluster with highest average feature value is taken as speech and rest clusters as non-speech. In some instances, where two prominent noise sources were present such as non-stationary background noise and occasional tonal impulsive noise, this approach led to three or more clusters.

D-SAD Decision Backend

- **FDK-SAD:** discriminative features (from PLTL data)
- **D-SAD:** CDF based linear curve
- Let $[FDK_{\{min\}}, 0]$ and $[FDK_{\{max\}}, 1]$ are end points of CDF curve
- We fit a straight line between them as

$$y = \frac{(x - FDK_{\{min\}})}{FDK_{\{max\}} - FDK_{\{min\}}}$$

- Now, we compute y for all x in range $[FDK_{\{min\}}, FDK_{\{max\}}]$.



- Point of intersection between straight line and CDF curves gives the SAD threshold.

[1] **H. Dubey**, and J. H. L. Hansen, "Speech Systems for Knowledge Extraction in Team Learning", (to be submitted at) *JASA Express Letters (2019).*

Proposed SAD Systems

- We combine proposed FDK-SAD features and three proposed backend namely VMGMM, DipSAD and D-SAD for comparisons

State-of-the-art		Proposed
System	**Feature**	**Decision Backend**
Combo-SAD (Sadjadi & Hansen, 2013)	Combo-SAD	2-GMM
Combo-VMGMM	Combo-SAD	VMGMM
FDK-VMGMM	FDK-SAD	VMGMM
Combo-DipSAD	Combo-SAD	DipSAD
FDK-DipSAD	FDK-SAD	DipSAD
FDK-DSAD	FDK-SAD	D-SAD
Combo-DSAD	FDK-SAD	D-SAD

SAD Baseline Systems

- **Combo-SAD Features (Sadjadi & Hansen, 2013) [6]: Unsupervised** Handcrafted five-dimensional features using harmonicity, clarity, prediction gain, periodicity and spectral flux. SAD threshold is given as convex combination of GMM means.
- **Semi-supervised GMM (SSGMM) (Sholokhov etal., 2018) [5]: Semi-supervised** It trains utterance-level speech-GMM and non-speech GMM. Higher energy frames takes as speech and lower energy frames as non-speech.
- **SohnSAD (Sohn etal., 1990) [3]: Unsupervised** Decision rule derived from generalized likelihood ratio test, Based on Frame-level energy
- **rSAD (Tan and Lindberg, 2010) [4]: Unsupervised** Selecting high-energy frames using aposteriori SNR. Frame-level pitch values computed. High-energy frames without a pitch value in reasonable range taken as non-speech.
- **USC-DNN-SAD (Van Segbroeck etal., *2013*) [2]: Supervised** 4-D features namely (i) spectral shape, (ii) spectro-temporal modulations, (iii) periodicity, and (iv) the long-term spectral variability are used for training a feedforward DNN for SAD.
- **TO-Combo-SAD [7] (Ziaei, Hansen etal., 2014) [6]: Semi-supervised** Combo-SAD features used to train a GMM for SAD using out-domain data. Later, trained GMM is used for decision making.
  - F0 ranges: male (85 -180 Hz), and female (65 -255 Hz)

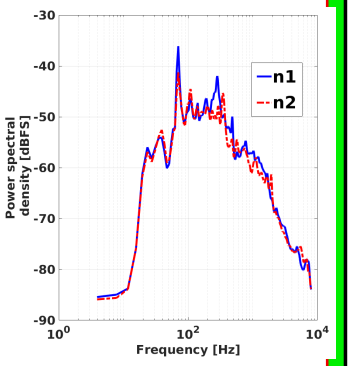
SAD Evaluation on CRSS-PLTL

- 5db SNR with Noise n1 and n2 CRSS-LDNN; "Overlapped-speech" and "Misc" included in feature extraction & decision making but excluded in scoring

Table: showing DCF (%) of all algorithms on PLTL data (approx. 80 minutes)

Systems	PLTL	PLTL+ n1	PLTL + n2
Combo-VMGMM	1.97	1.99	2.35
FDK-VMGMM	2.01	2.16	2.17
Combo-DipSAD	2.84	2.96	2.76
FDK-DipSAD	7.23	7.50	7.12
FDK-DSAD	15.29	N/A	N/A
Combo-DSAD	17.68	N/A	N/A
SohnSAD [3]	28.20	28.51	28.71
rSAD [4]	49.57	49.57	49.65
SSGMM [5]	28.95	29.13	30.58
TO-Combo-SAD [7]	29.16	N/A	N/A
USC-DNN-SAD [2]:	23.19	N/A	N/A

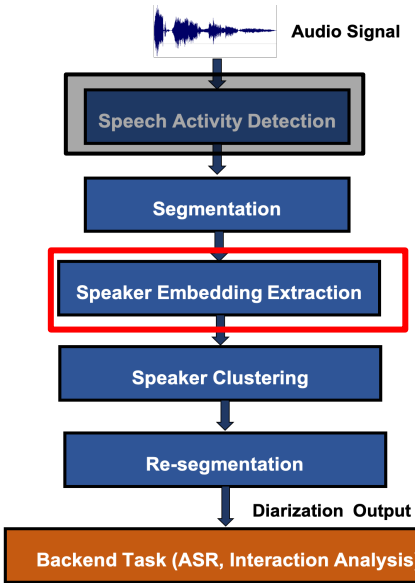
$$DCF = 0.5 * P_{\{miss\}} + 0.5 * P_{\{fa\}}$$



SAD Research: Summary & Future

- **FDK-SAD:** Discriminative features
- **VMGMM:** Flexible model for decision making, semi-supervised
- **Dip-SAD:** Non-parametric approach for low-resource situations
- **D-SAD:** Computationally simple approach
- Proposed methods lead to significant reductions in DCF for naturalistic CRSS-PLTL and NIST corpora (OpenSAD 2015 and OpenSAT 2017).
- Two phase of evaluations (1) standalone SAD assessment; (2) SAD for text-dependent speaker verification on RedDots
- Significant reverberation, overlap and multiple noise sources break state-of-art methods: SohnSAD, SSGMM and rSAD broke as their performance depends on speech energy and fundamental frequency (F0)
- Getting accurate F0 for naturalistic data is challenging
- **Future Work:**
  - Use FDK for ASR
  - Introducing slope parameter into D-SAD

Where are we now?



Sinc Neural Nodes

- Infusing DSP knowledge into Neural Nets
- Band-pass spectral filtering ➡ Sinc functions as convolutive filters
- Sinc Layer: Each node perform convolution with Sinc functions
- Neural architecture for processing time-domain raw waveform

$$sinc(x) = sin(x)/x$$

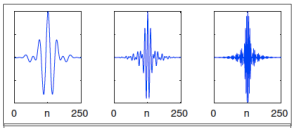
$$H[f, f_1, f_2] = rect\left(\frac{f}{2f_2}\right) - rect\left(\frac{f}{2f_1}\right)$$

$$h[m, f_1, f_2] = 2f_2 \cdot sinc(2\pi f_2 m) - 2f_1 \cdot sinc(2\pi f_1 m)$$

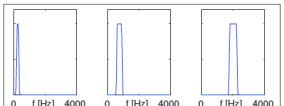
$$f_1' = |f_1|$$
$$f_2' = |f_1| + |f_2 - f_1|$$

$$h_w[m, f_1', f_2'] = h[m, f_1', f_2'] \cdot w_{hamming}[m]$$

$$w_{hamming}[m] = 0.54 - 0.46 \cdot cos\left(\frac{2\pi m}{L}\right)$$



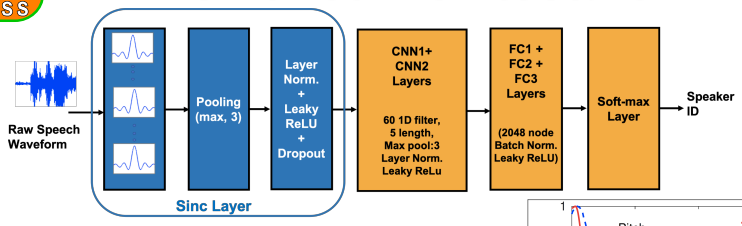
Time-domain: Sinc filter



Frequency response:Sinc filter

[1] **H. Dubey**, A. Sangwan and J. H. L. Hansen, "Transfer Learning Using Raw Waveform SincNet For Robust Speaker Diarization", *IEEE ICASSP 2019*, May 12-19, Brighton, UK.
[2] M. Ravanelli, Y. Bengio. "Speaker Recognition from raw waveform with SincNet", *IEEE SLT 2018*, Athens, Greece, Dec. 2018.

SincNet Architecture



- Input: Raw waveform SincNet
- Sinc Layer: 80 sinc filters (length 251), max-pooling (3).
- Compact architecture: Total 2*80 parameters
- Next, two 1-D CNN layers and three fully connected (FC) layers; Final soft-max layer perform speaker classification.
- Faster convergence compared to standard CNN

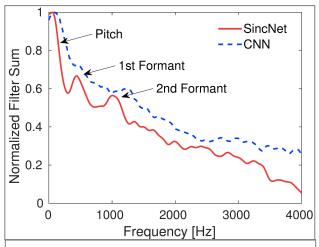
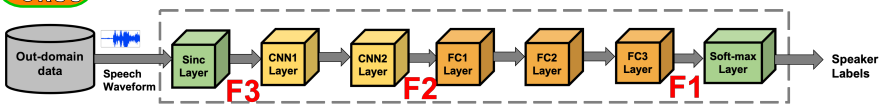


Fig. 2: Cumulative frequency response of SincNet and CNN filters on speaker-id (Librispeech).

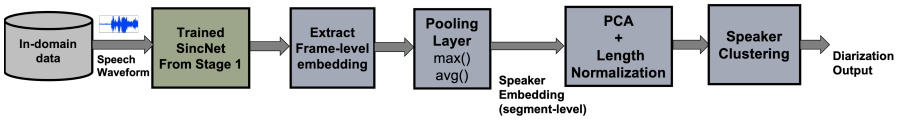
- Sinc-Layer learn the formants and pitch trajectory
- Leads to efficient speaker modeling in compact representations

[1] M. Ravanelli, Y. Bengio. "Speaker Recognition from raw waveform with SincNet", *IEEE SLT 2018*, Athens, Greece, Dec. 2018.

SincNet for Diarization



(a) Stage 1: Training SincNet for Frame-level Speaker Recognition using Out-domain data.

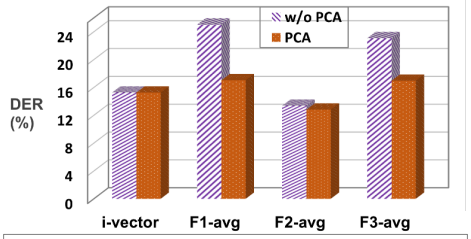


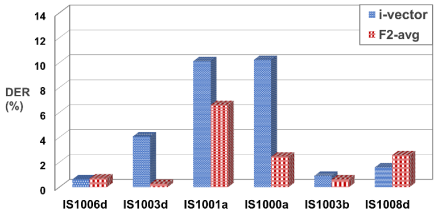
(b) Stage 2: Vanilla Transfer Learning (VTL) based on pre-trained SincNet extracts speaker embeddings for Speaker Diarization of In-domain data.

- Unsupervised Transfer Learning implemented in two stages:
- Stage 1: SincNet is trained for frame-level (10ms) speaker ID on out-domain data.
- Stage 2: Trained SincNet is used as feature extractor for in-domain data. **Frame-level embeddings** are max() or avg() pooled to get utterance/segments-level embeddings.
- Next, PCA, length-normalization and Speaker Clustering follows.

[1] **H. Dubey**, A. Sangwan and J. H. L. Hansen, "Transfer Learning Using Raw Waveform SincNet For Robust Speaker Diarization", *IEEE ICASSP 2019*, May 12-19, Brighton, UK.
[2] C. Zhang, S. Ranjan, J.H.L. Hansen, "'An Analysis of Transfer Learning for Domain Mismatched Text-independent Speaker Verification," *ISCA Speaker Odyssey-2018*, June 26-29, Les Sables d'Olonne, France.

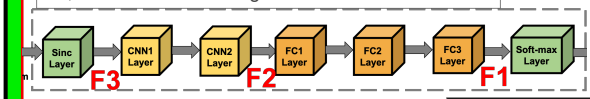
SincNets Results: DER





PLTL data: Comparing i-vector with average pooled F1, F2 and F3 embeddings. w/o PCA= No PCA

AMI corpus: F2-avg with PCA (51 dim.) shows significant improvements over i-vector with PCA.



PLTL data corpus: Effect of PCA (51 dim.) on DER

Results w/ PLTL data: **(F2-avg)**

	i-vector	F2-avg	F2-max
w/o PCA	15.26	13.37	43.55
PCA	15.26	**12.81**	14.36

- F2-avg gives Best (least) DER among all alternatives here
- PCA helps in improving Diarization performance

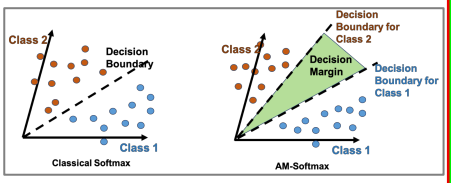
Additive Margin (AM) - Softmax

- Softmax loss:
  - Optimizes the inter-class variance;
  - Lacks the capacity to reduce the intra-class variations.

$$L_{softmax} = -\frac{1}{n}\sum_{i=1}^{n}\log\left\{\frac{e^{||W_{y_i}||\cdot||f_i||\cdot\cos(\theta_{y_i})}}{\sum_{j=1}^{C} e^{||W_j||\cdot||f_i||\cdot\cos(\theta_j)}}\right\}$$

$$L_{ams} = -\frac{1}{n}\sum_{i=1}^{n}\log\left\{\frac{e^{s\cdot(W_{y_i}^T\cdot f_i - m)}}{e^{s\cdot(W_{y_i}^T\cdot f_i - m)} + \sum_{j=1, j\neq y_i}^{C} e^{s\cdot W_{y_i}^T\cdot f_i}}\right\}$$

Margin parameter



Graphical illustration of conventional Softmax and AM-Softmax

- AM-Softmax (Additive Margin):
  - minimize intra-class variance.
  - Introduce a margin around decision boundary in Softmax
  - Large margin further reduce the intraclass variance

[1] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters, (2018),* vol. 25, no. 7, pp. 926–930.

Center Loss (CL)

- CL: Supervised loss used jointly with Softmax/AM-Softmax loss
- Simultaneously, (i) learns a center for deep embeddings of each speaker; and (ii) penalizes the distances between deep features and their corresponding class centers.
- **CL + Softmax has two benefits:**
  - Inter-class dispension ↑(Softmax),
  - Intra-class compactness (CL).

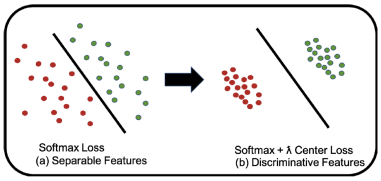


Illustration: CL enhances the discriminating power of deep speaker embeddings

Speaker Embedding

$$L_{center} = \frac{1}{2}\sum_{i=1}^{n} \|x_i - c_{y_i}\|_2^2$$

Centroid of corresponding class

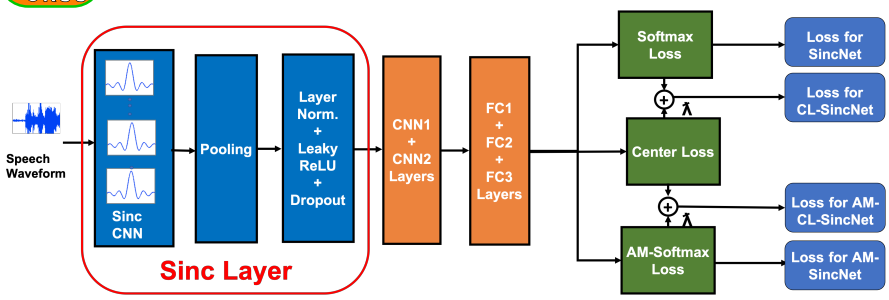
$$L_{total} = L_{softmax} + \lambda L_{center}$$

Balancing parameter, small value < 1

- CL aims to improve the discriminative power of SincNet embeddings:
  - By minimizing the intra-class variance, and
  - Ensuring the separability of different classes.

[1] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision, (2016),* pp. 499–515.

Proposed Architecture



- Bringing the discrimination power of CL and AM-Softmax into SincNets
- Proposed:
  - CL-SincNet, AM-CL-SincNet for Speaker ID
  - All four architectures for Speaker Diarization
- Trained on Librispeech (2484 speakers) and TIMIT (462 speakers)

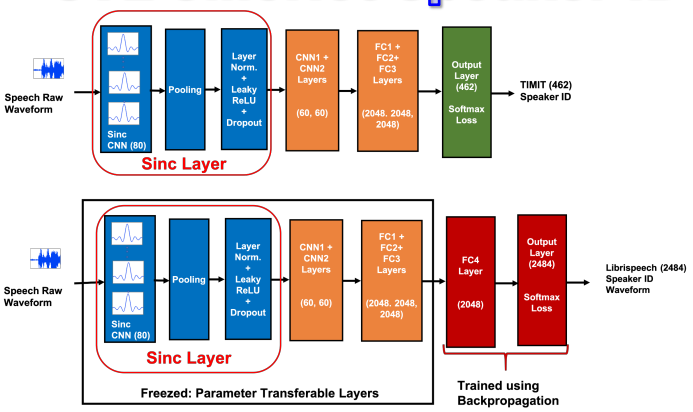
[1] **H. Dubey**, A. Sangwan and J. H. L. Hansen, "Transfer Learning Using Raw Waveform SincNet For Robust Speaker Diarization", *IEEE ICASSP 2019*, May 12-19, Brighton, UK.
[2] **H. Dubey**, and J. H. L. Hansen, "SincNet Speaker Embeddings and Model-based Clustering for Speaker Diarization in Naturalistic Audio Streams", (to be submitted at) *IEEE/ACM TASLP (2019)*.

STL-SincNet

- **Supervised Transfer Learning (STL):** use of additional source of information (from out-domain source task) into an supervised classification on target domain.
- **Goal:**
  - Improve learning in the target task by leveraging knowledge from the source task.
  - STL-SincNet transfer speaker discriminative capability from TIMIT to Librispeech
- **Improvements obtained:**
  - Better initial performance
  - Faster speed of convergence
  - better final performance
  - **Better unsupervised transfer to Diarization pipeline**

[1] **H. Dubey**, and J. H. L. Hansen, "SincNet Speaker Embeddings and Model-based Clustering for Speaker Diarization in Naturalistic Audio Streams", (to be submitted at) *IEEE/ACM TASLP (2019).*

STL-SincNet Speaker ID



- **Motivation:** Early layers in SincNet learn robust, domain-invariant features and later layers learn speaker discrimination for speakers contained in training data
- Fine-tune the pre-trained SincNet on TIMIT w/ Groundtruth SAD.

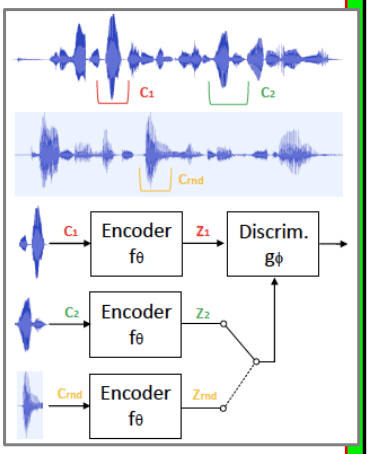
[1] **H. Dubey**, and J.H.L. Hansen, "SincNet Speaker Embeddings and Model-based Clustering for Speaker Diarization in Naturalistic Audio Streams", (to be submitted at) *IEEE/ACM TASLP (2019).*

Mutual Information (MI)-SincNet

- Mutual Information (MI)-SincNet [1]: Leverages MI to learn speaker embeddings in an unsupervised setup.

$$MI(z_1, z_2) = \int_{z_1}\int_{z_2} p(z_1, z_2) log\left(\frac{p(z_1, z_2)}{p(z_1)p(z_2)}\right) dz_1 dz_2$$
$$= D_{KL}\big(p(z_1, z_2)||p(z_1)p(z_2)\big),$$

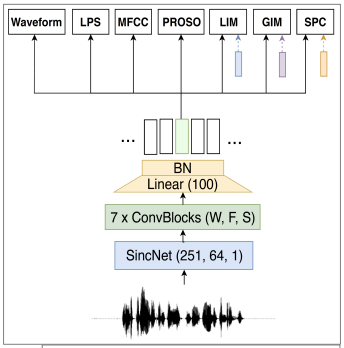
- Implicit optimization of MI is achieved via Encoder-discriminator architecture.
- SincNet encoder: transforms raw speech to bottleneck features.
- Discriminator trained to separate positive samples (codes from same speakers/sentence) or negative samples (from different speakers).
- Maximizing MI between the encoded representations of speech chunks from same speaker.
- Discriminator trained using the standard binary cross-entropy (BCE) loss: $L(\Theta, \Phi) = \mathbb{E}_{X_p}[\log(g(z_1, z_2))] + \mathbb{E}_{X_n}[\log(1 - g(z_1, z_{rnd}))]$



[1] Ravanelli, Mirco, and Yoshua Bengio. "Learning Speaker Representations with Mutual Information." *arXiv preprint* arXiv:1812.00271 (*2018*).

Proposed MI-SincNet for Diarization

- PASE [1] = Problem-Agnostic Speech Encoder
- PASE tasks: Waveform, Log power spectrum (LPS), Mel-frequency cepstral coefficients (MFCC), Prosody, Local info max (LIM), Global info max (GIM), Sequence predicting coding (SPC)
- Proposed MI-SincNet for Diarization uses PASE architecture but trains it on MI based speaker identification and do not include PASE tasks.
- Bottleneck feature of 100 dim. are extracted from each 200ms frames with 10ms skip.
- Avg() pooling in applied on frame-level speaker embeddings to get segment-level embeddings for Diarization



The PASE architecture [1], with the considered tasks.

[1] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio. "Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks." *arXiv preprint* arXiv:1904.03416 (*2019*).

SincNet : Impact of SAD

- **SAD impact on SincNet training**

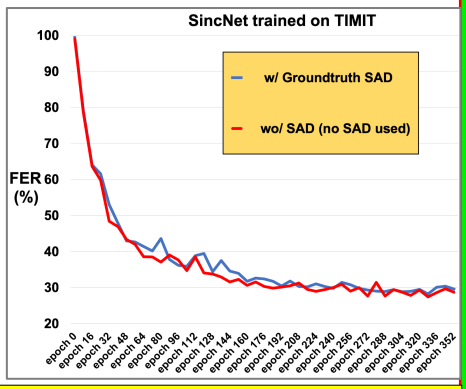
Option1: Ground-truth SAD used train SincNet

Option2: No SAD used in training SincNet

- Speaker ID Frame error rate (FER) remains similar towards end of training/convergence
- **PLTL DER% (Best)**

wo/ SAD, TIMIT trained model: **15.42%**

w/ Groundtruth SAD,
TIMIT trained model: **10.63%**



- SincNet robust to presence of pause/silence for in-domain speaker ID
- Presence of SAD has limited impact on SID Frame-error-rate; However, effective SAD (groundtruth) has major impact on final DER%!
- SincNet w/ SAD generalizes better on out-domain Diarization task

[1] **H. Dubey**, and J. H. L. Hansen, "SincNet Speaker Embeddings and Model-based Clustering for Speaker Diarization in Naturalistic Audio Streams", (to be submitted at) *IEEE/ACM TASLP (2019).*

AM-SincNet : Impact of SAD

- **SAD impact on AM-SincNet training**
- Trained with Margin Weight of m=0.90 in two cases:

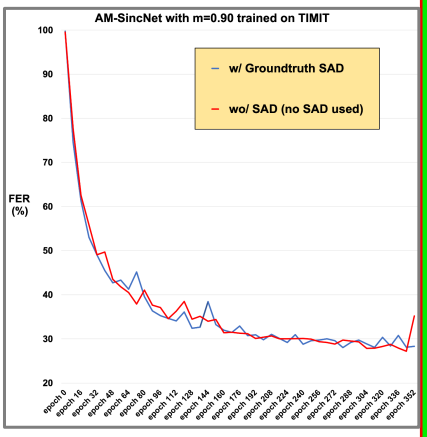
Option1: Ground-truth SAD used train AM-SincNet

Option2: No SAD used in training AM-SincNet

- Good news: Frame error rate (FER) remains similar towards end of training/convergence
- **PLTL DER (%) (Best)**

wo/ SAD, TIMIT trained model: **8.04 %**

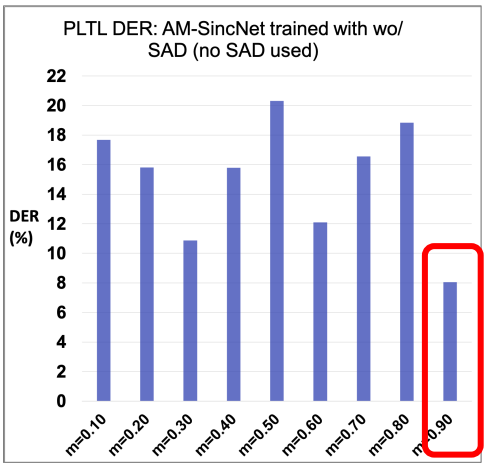
w/ Groundtruth SAD,
TIMIT trained model: **12.10 %**



- AM-SincNet is robust to presence of silences similar to SincNet
- Ground-truth SAD more stable training; Margin need to be tuned.
- In noiser data and wo/ SAD, DER is better than SincNet (AM helps)

PLTL Results: DER for AM-SincNet

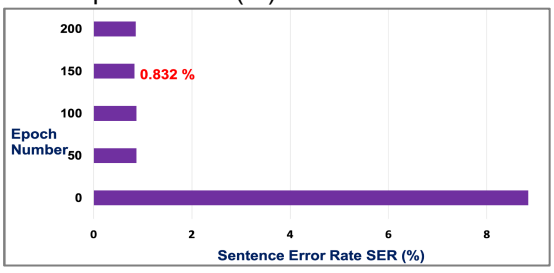
- Unsupervised Transfer Learning (UTL)
- Trained wo/SAD (no SAD used)
- Trained with Different margins (m)
- Trained on TIMIT data (training set)
- Cosine K-means clustering
- **UTL is complex process**



- F2-avg from AM-SincNet: PLTL DER % (Best) for margin = 0.90
- Higher margin leads to robustness in unsupervised Transfer Learning

STL-SincNet Results

Librispeech SER (%) for STL-SincNet



System	Best PLTL DER (%)
SincNet	10.63
AM-SincNet	**8.04** (m= 0.90)
CL-SincNet	14.81
AM-CL-SincNet	15.01 (m=0.5, CL=0.5)
TL-SincNet (F1avg)	13.29
MI-SincNet	15.71

- Early layers in SincNet learn robust, domain-invariant features
- Later layers learn speaker discrimination for training set
- STL-SincNet Fine-tune pre-trained SincNet on TIMIT w/ grount-truth SAD
- Tabular results are from conducted experiments not all possible ones

CL-SincNet: TIMIT Speaker ID

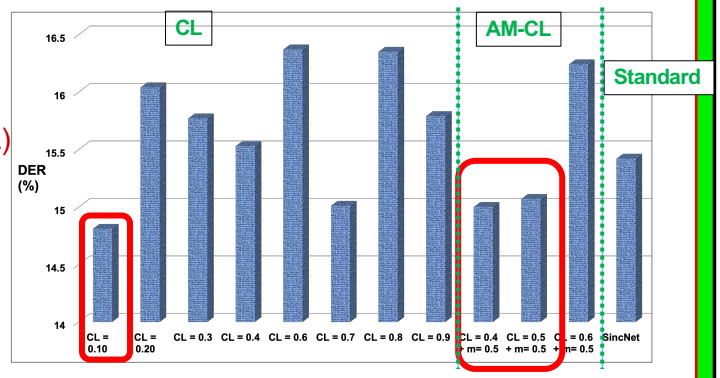
- Sentence Error Rate for Speaker ID using CL-SincNet

CL parameter	SER(%)
0.10	0.3608
0.20	0.2886
0.30	0.2165
0.40	0.3608
0.50 + AM m=0.5	0.2886
0.60	0.2886
0.70	0.2886
0.80	0.1443
0.90	0.2886
SincNet	0.7937
SincNet w/ ground-truth SAD	0.5772

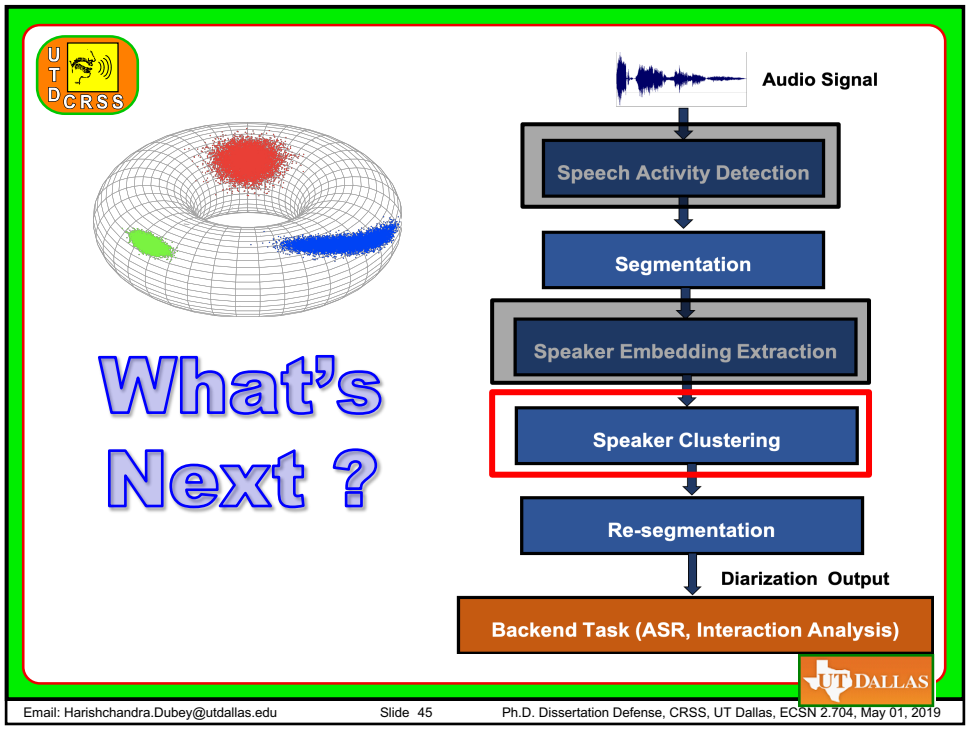
CL-SincNet always better than standard SincNet in terms of SER (%)

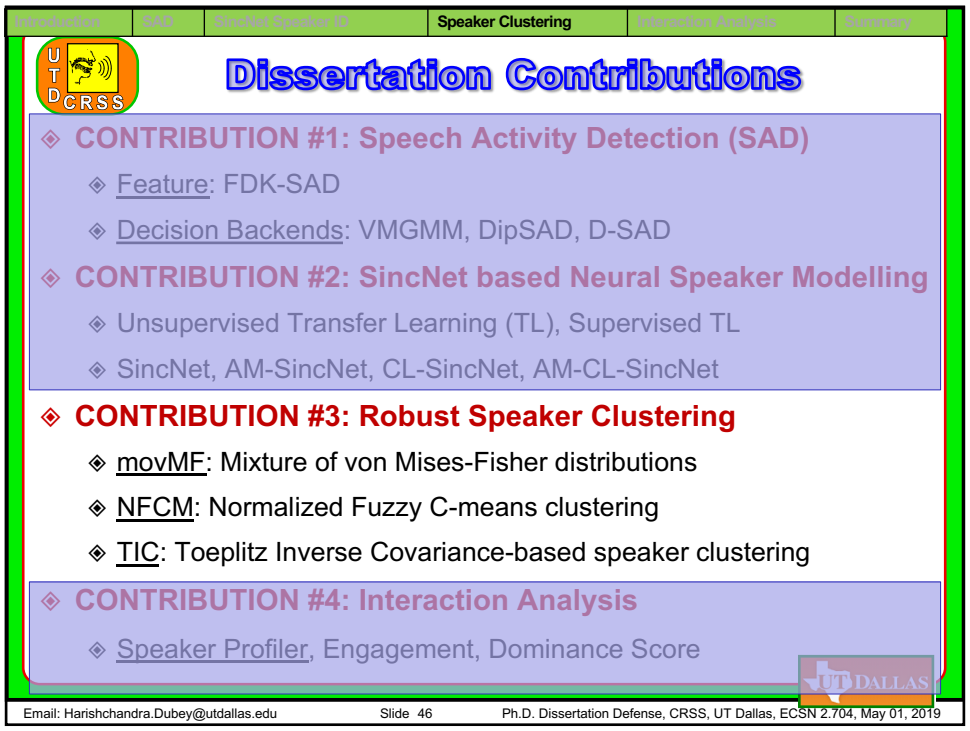
PLTL DER: CL-SincNets

- Unsupervised Transfer Learning (UTL)
- Trained wo/SAD (no SAD used)

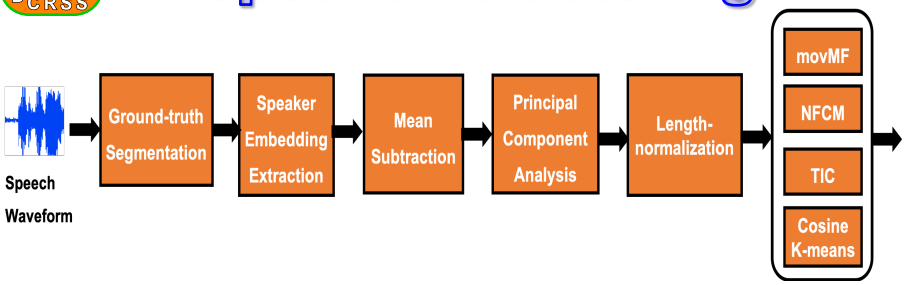


- F2-avg from CL-SincNet: PLTL DER % (Best) for CL (lambda) = 0.10
- Lower CL params. leads to robustness in Diarization (UTL)
- AM-CL-SincNet: Additive Margin (AM) improves generalization in UTL and hence better DER for CL-SincNets
- Careful selection of CL and AM parameters is important


Audio Signal
Speech Activity Detection
Segmentation
Speaker Embedding Extraction
Speaker Clustering
Re-segmentation
Diarization Output
Backend Task (ASR, Interaction Analysis)
What's Next ?
UTDALLAS
Email: Harishchandra.Dubey@utdallas.edu
Slide 45
Ph.D. Dissertation Defense, CRSS, UT Dallas, ECSN 2.704, May 01, 2019


Introduction
DAD
SincNet Speaker ID
Speaker Clustering
Interaction Analysis
Summary
Dissertation Contributions
CONTRIBUTION #1: Speech Activity Detection (SAD)
Feature: FDK-SAD
Decision Backends: VMGMM, DipSAD, D-SAD
CONTRIBUTION #2: SincNet based Neural Speaker Modelling
Unsupervised Transfer Learning (TL), Supervised TL
SincNet, AM-SincNet, CL-SincNet, AM-CL-SincNet
CONTRIBUTION #3: Robust Speaker Clustering
movMF: Mixture of von Mises-Fisher distributions
NFCM: Normalized Fuzzy C-means clustering
TIC: Toeplitz Inverse Covariance-based speaker clustering
CONTRIBUTION #4: Interaction Analysis
Speaker Profiler, Engagement, Dominance Score
UTDALLAS
Email: Harishchandra.Dubey@utdallas.edu
Slide 46
Ph.D. Dissertation Defense, CRSS, UT Dallas, ECSN 2.704, May 01, 2019

Speaker Clustering



- Evaluation pipeline includes ground-truth segmentation
- Except TIC all algorithms require length-normalization
- Mean subtraction requires mean from entire meeting
- Speaker embeddings extracted from entire segment

[1] **H. Dubey**, A. Sangwan and J. H. L. Hansen, "Robust Speaker Clustering using Mixtures of von Mises-Fisher Distributions for Naturalistic Audio Streams", *ISCA INTERSPEECH, (2018)*, Hyderabad, India, Sept. 2-6, 2018.
[2] **H. Dubey**, A. Sangwan, J.H.L. Hansen, "Toeplitz Inverse Covariance based Robust Speaker Clustering for Naturalistic Audio Streams," *(submitted to) ISCA INTERSPEECH, 2019.*
[3] **H. Dubey**, and J. H. L. Hansen, "SincNet Speaker Embeddings and Model-based Clustering for Speaker Diarization in Naturalistic Audio Streams", (to be submitted at) *IEEE/ACM TASLP (2019).*

movMF Speaker Clustering

- Motivation: von Mises-Fisher (vMF) distributions model data on unit hypersphere such as L2-normalized speaker embeddings
- Mixture Model movMF: One component per speaker; Expectation Maximization (EM) for estimation of movMF model

$$f(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{h=1}^{N_c} \alpha_h f_h(\boldsymbol{x}|\boldsymbol{\theta_h})$$

Weights of each component

$$f_h(\boldsymbol{x}|\boldsymbol{\mu}, \kappa) = c_d(\kappa) e^{\kappa \boldsymbol{\mu}^T \boldsymbol{x}}$$

Mean vector

$$c_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)}$$

Concentration parameter

modified Bessel function of first-kind and order (d/2)-1

movMF Speaker Clustering

- **In E-step**: embeddings are assigned to clusters such that the likelihood is maximized.
- **In M-step:** model parameters are updated

- movMF model do not have closed form solutions, we approximate!
- EM is used for hard clustering where each data point is assigned to one unique cluster in E-step.

Input: (1) Set χ of n segment-level normalized i-Vectors of dimensions d from complete audio recording; (2) Number of clusters, N_c.
Output: (1) A disjoint partitioning of χ into N_c clusters; (2) Model parameters of mixture of N_c d-variate vMF distributions.
METHOD:

Initialize all α_h, μ_h, κ_h for $h = 1, \cdots, N_c$
Repeat
{The hardened Expectation-step of EM}
for $i = 1$ to n **do**
  for $h = 1$ to N_c **do**
    $f_h(\chi_i|\theta_h) \leftarrow c_d(\kappa_h)e^{\kappa_h \mu_h^T \chi_i}$
  end-for
  for $h = 1$ to N_c **do**
  The hardened-distribution of hidden-variables is given by

$$q(h|\chi_i, \Theta) \leftarrow \begin{cases} 1, & \text{if } h = \arg\max_{h'} \alpha_{h'} f_{h'}(\chi_i|\theta_{h'}) \\ 0, & \text{otherwise} \end{cases}$$

  end-for
end-for
{The Maximization-step of EM}
  for $h = 1$ to N_c **do**
    $\alpha_h \leftarrow \frac{1}{n}\sum_{i=1}^{n} q(h|\chi_i, \Theta)$
    $\mu_h \leftarrow \sum_{i=1}^{n} \chi_i \, q(h|\chi_i, \Theta)$
    $\bar{r} \leftarrow \frac{||\mu_h||}{n\alpha_h}$
    $\mu_h \leftarrow \frac{\mu_h}{||\mu_h||}$
    $\kappa_h \leftarrow \frac{\bar{r}d - \bar{r}^3}{1 - \bar{r}^2}$
  end-for
Until *convergence*

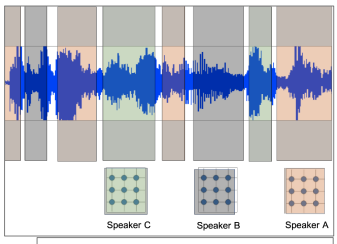
NFCM Clustering

- NFCM: Normalized Fuzzy C-means is an extension of fuzzy c-means for length-normalized data
- **Principle:** Each feature vector can belong to more than one cluster with different membership values in range [0,1].
- **Conditions for minimization of Least-Squared Error**
  - Membership values lies in range [0,1].
  - Membership values for each feature vector must sum to 1.
  - The sum of all membership values in a speaker cluster must be smaller than the number of data points.

- Angular difference is used as similarity measure
- Distribution-free approach
- low computational time

TIC Speaker Clustering

- TIC: Toeplitz Inverse Covariance
- Motivation: Small set of speakers (~10) repeat throughout the audio stream
- Highlights: Segment-level speaker features are modelled as MRF correlation network where conversational turns are a sequence of states.
- Speaker MRFs capture the time-invariant partial correlation structure present in all speech segments belonging to that speaker.

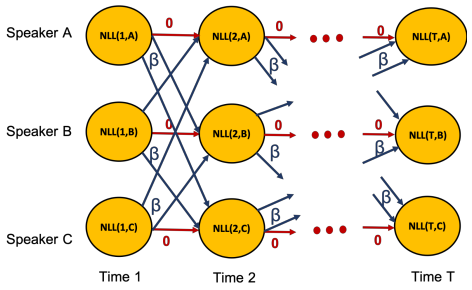


Each speaker A, B or C is characterized by its correlation network defined as a Markov Random Field (MRF).

- Implemented as two Algorithms
- Algorithm 1: Assign-Clusters
- Algorithm 2: EM based iterations for speaker clustering

[1] **H. Dubey**, A. Sangwan, J.H.L. Hansen, "Toeplitz Inverse Covariance based Robust Speaker Clustering for Naturalistic Audio Streams," *(submitted to) ISCA INTERSPEECH, 2019.*

TIC Speaker Clustering



- Approach: Viterbi-like approach assigns the clusters in a way such that
- The Likelihood of data given the model is maximized
- State transition between successive segments in minimized where \beta is switch penalty

Algorithm 3 TIC-1: Assign-Clusters

Input: GIVEN $\beta \geq 0$, $-LogL(i,j)$ = negative log-likelihood for i-th feature vector when it is assigned to j-th speaker cluster. K is the number of speakers. Time stamp of i-vectors (speaker features) is from 1 to T.

Output: FinalPath i.e. cluster assignment for each i-vector.

METHOD:

```
Initialize
    previous_cost = list of K zeros
    current_cost = list of K zeros
    previous_path = list of K empty lists
    current_path = list of K empty lists

for i = 1, . . . , T do
  for j = 1, . . . , K do
    min_index = index of minimum {previous_cost}
    if previous_cost[min_index] + β > previous_cost[j]
then
      current_cost[j] = previous_cost[j] −LogL(i, j)
      current_path[j] = previous_path[j].append[j]  else
       current_cost[j] = previous_cost[minIndex] + β
−LogL(i, j)
      current_path[j]=previous_path[min_index].append[j]
  previous_cost = current_cost
  previous_path = current_path
final_min_index = index of minimum {current_cost }
FinalPath = current_path[final_min_index]
return FinalPath
```

TIC Speaker Clustering

- **In E-step**: embeddings are assigned to clusters [Viterbi path with least cost]

$$\text{minimize} \sum_{i=1}^{K} \sum_{X_t \in C_i} N(X_t, \Theta_i) + \beta 1_{X_{t-1} \notin C_i}$$

- **In M-step:** model parameters are updated using dynamic programming (DP) and alternating direction method of multipliers (ADMM) [Toeplitz Graphical Lasso]

$$\arg \min_{\Theta \in \mathbb{T}, \mathbb{C}} \sum_{i=1}^{K} \left[||\lambda \circ \Theta_i||_1 + \sum_{X_t \in C_i} \left(N(X_t, \Theta_i) + \beta 1_{X_{t-1} \notin C_i} \right) \right.$$

Algorithm 4 TIC-2: Proposed TIC-based Speaker Clustering

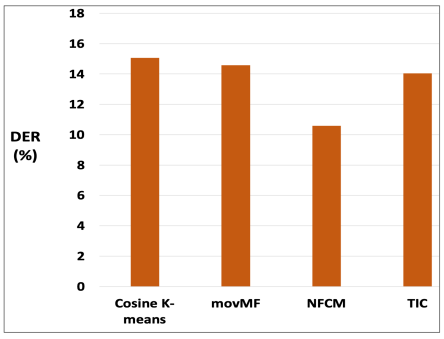
Input: GIVEN: (i) Algorithm 3 for assigning i-vectors to speaker clusters; (ii) i-Vectors (features) for time 1 to T.

METHOD:

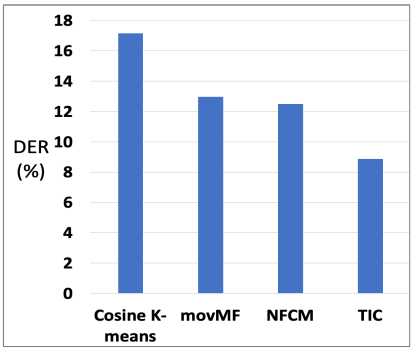
1: **Initialize**
2: Speaker cluster parameters, Θ
3: Diarization output, spk_labels = cluster assignment set $\mathbb{C}$
4: **Repeat**
5:
6: **E-step:** Assign feature vectors to speaker clusters using Algorithm 3 i.e., map i-vectors $\Rightarrow$ spk_labels (see section 3.1.)
7:
8: **M-step:** Update speaker cluster (model) parameters by solving the Toeplitz Graphical Lasso (see section 3.2.) i.e., $spk\_labels \Rightarrow \Theta$
9:
10: **Until** Stationarity.
11: **return** (Θ, spk_labels)

- Beta is switch penalty and ensure smoothness in decoded path
- Lambda is sparsity constraints imposed on Inverse Covariance Matrix

Clustering Results: PLTL

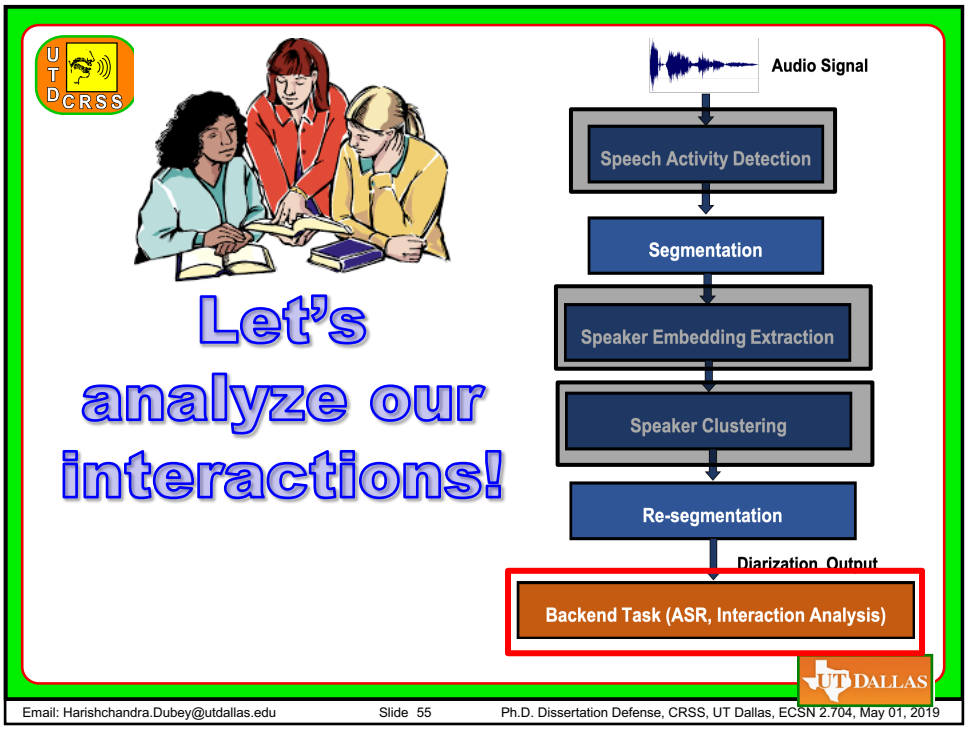


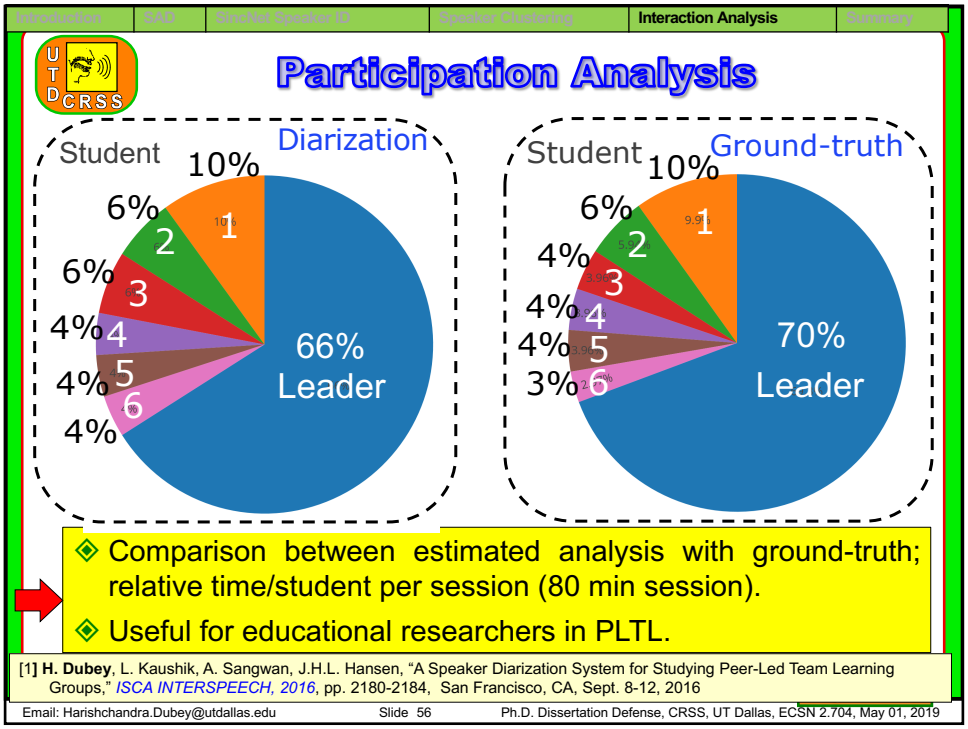
(a) PLTL: F2-avg AM-SincNet m=0.95



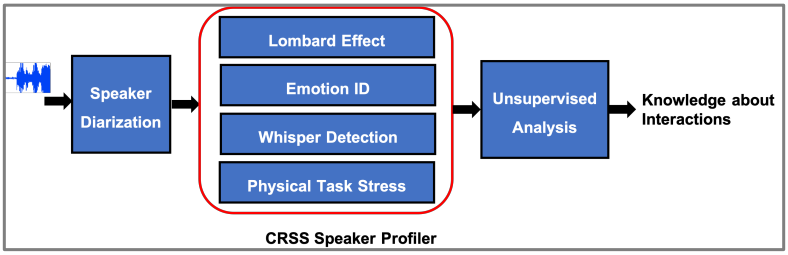
(b) PLTL: i-vector

- TIC has the BEST performance without PCA as it works on correlation!
- TIC achieves significant reduction in DER as compared to other methods
- TIC gets less than 1% DER for AMI meetings
- When SincNet embeddings are used in Cosine K-means, DER significant reduced compared to i-vectors.

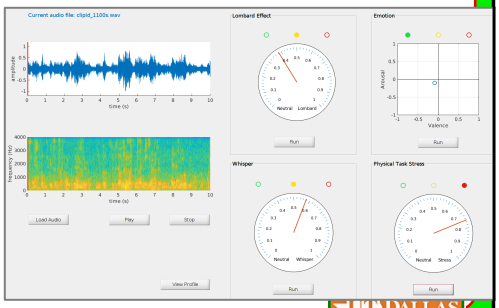

Let's analyze our interactions!
Audio Signal
Speech Activity Detection
Segmentation
Speaker Embedding Extraction
Speaker Clustering
Re-segmentation
Diarization Output
Backend Task (ASR, Interaction Analysis)
UT DALLAS
Email: Harishchandra.Dubey@utdallas.edu
Slide 55
Ph.D. Dissertation Defense, CRSS, UT Dallas, ECSN 2.704, May 01, 2019


Introduction
DAD
Enrolled Speaker ID
Speaker Clustering
Interaction Analysis
Summary
Participation Analysis
Diarization
Student
10%
6%
6%
4%
4%
4%
66%
Leader
Ground-truth
Student
10%
6%
4%
4%
4%
3%
70%
Leader
Comparison between estimated analysis with ground-truth; relative time/student per session (80 min session).
Useful for educational researchers in PLTL.
[1] H. Dubey, L. Kaushik, A. Sangwan, J.H.L. Hansen, "A Speaker Diarization System for Studying Peer-Led Team Learning Groups," ISCA INTERSPEECH, 2016, pp. 2180-2184, San Francisco, CA, Sept. 8-12, 2016
Email: Harishchandra.Dubey@utdallas.edu
Slide 56
Ph.D. Dissertation Defense, CRSS, UT Dallas, ECSN 2.704, May 01, 2019

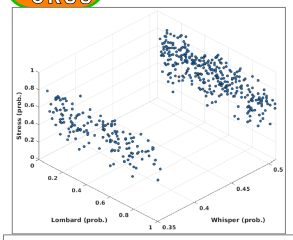
CRSS Speaker Profiler



- 4 Low-level Features: Emotion, Whisper, Physical Task Stress and Lombard effect.
- Motivation: Use these features to detect engagement and communication behaviors
- Speech profiler was used to get the probabilities of Lombard effect, whisper and physical stress.

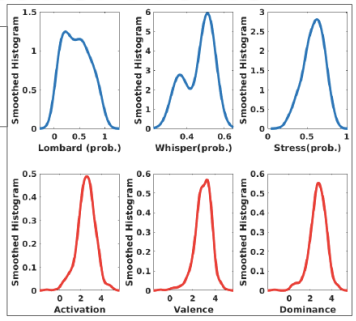


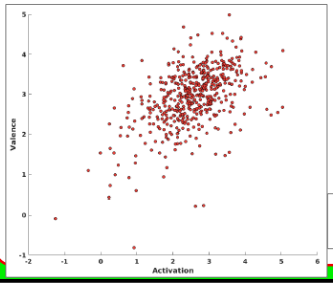
Results: CRSS Speaker Profiler



Shows 3D scatter plot for physical task profile (PTP). It corresponds to 80 minute PLTL session with 7 students and peer-lead.

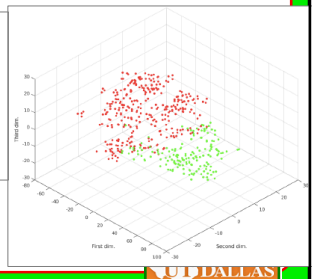
Showing smoothed density for all six features. Only Whisper is bi-modal



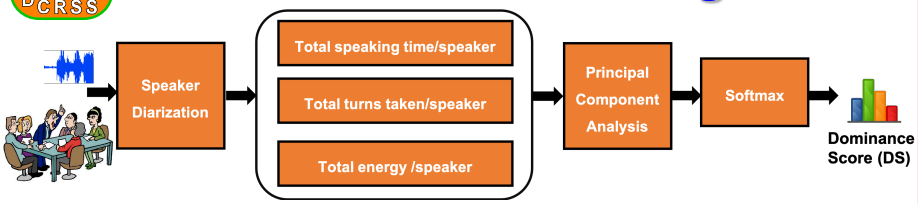


2D scatter plot showing valence and activations

t-SNE plots with 3-D embedding of mean normalized 6-dimensional features.Color coding shows two clusters from K-means. t-SNE used Euclidean distances for this plot.



Dominance Modelling



- **Idea**: Dominance score (DS) is estimated using unsupervised analysis of each speaker segment.
- **Approach**: Dominance is correlated with speaking time and turns taken by the speakers [2]. We added speaker energy as third parameter.
- Diarization makes this process unsupervised and automatic.
- Pearson's correlation between DS and ground-truth dominance rating is 0.8748.
- **Ground-truth:** is average of scores from three annotators

[1] **H. Dubey**, A. Sangwan and J. H. L. Hansen, "A Robust Diarization System for Measuring Dominance in Peer-Led Team Learning Groups", *IEEE SLT, (2016)*, San Diego, USA, pp. 319–323.
[2] M. Schmid Mast, "Dominance as expressed and inferred through speaking time," *Human Communication Research, (2002)*, vol. 28, no. 3, pp. 420–450, 2002.

Dissertation Contributions

- **CONTRIBUTION #1: Speech Activity Detection (SAD)**
  - Feature: FDK-SAD
  - Decision Backends: VMGMM, DipSAD, D-SAD
- **CONTRIBUTION #2: SincNet based Neural Speaker Modelling**
  - Unsupervised Transfer Learning (TL), Supervised TL
  - SincNet, AM-SincNet, CL-SincNet, AM-CL-SincNet
- **CONTRIBUTION #3: Robust Speaker Clustering**
  - movMF: Mixture of von Mises-Fisher distributions
  - NFCM: Normalized Fuzzy C-means clustering
  - TIC: Toeplitz Inverse Covariance-based speaker clustering
- **CONTRIBUTION #4: Interaction Analysis**
  - Speaker Profiler, Engagement, Dominance Score

List of Publications

[JP1] **H. Dubey** and J. H.L. Hansen, " SincNet Speaker Embeddings and Model-based Clustering for Speaker Diarization in Naturalistic Audio Streams", (to be submitted at) *IEEE/ACM TASLP, (2019).*

[JP2] H. Dubey, A. Sangwan and J. H.L. Hansen, "Leveraging Frequency-Dependent Kernel and DIP-Based Clustering for Robust Speech Activity Detection in Naturalistic Audio Streams", *IEEE/ACM Trans. on Audio, Speech, and Language Processing, (2018),* 26.11 (2018): 2056-2071.

[JP3] H. Dubey, A. Sangwan and J. H.L. Hansen, "Using Speech Technology for Quantifying Behavioral Characteristics in Peer-Led Team Learning Sessions", *Computer Speech & Language, (2017), 46 :* 343-366.

[JP4] H. Dubey and J. H.L. Hansen, "Speech System for Knowledge Extraction in Team Learning", (to be submitted at) *JASA Express Letters, (2019).*

[CP1] H. Dubey, A. Sangwan and J. H. L. Hansen, "Toeplitz Inverse Covariance based Robust Speaker Clustering for Naturalistic Audio Streams", (under review) *ISCA INTERSPEECH, (2019)*, Graz, Austria, Sept. 15-19.

[CP2] H. Dubey, A. Sangwan and J. H. L. Hansen, "Transfer Learning using Raw Waveform SincNet for Robust Speaker Diarization", *IEEE ICASSP, (2019)*, Brighton, UK, 12-17 May.

[CP3] C. Zhang, F. Bahmaninezhad, S. Ranjan, **H. Dubey**, W. Xia, John H. L. Hansen, "UTD-CRSS Systems for 2018 NIST Speaker Recognition Evaluation", *IEEE ICASSP (2019)*, Brighton, UK, 12-17 May.

[CP4] H. Dubey, A. Sangwan, J. H. L. Hansen, "Robust Speaker Clustering using Mixtures of von Mises-Fisher Distributions for Naturalistic Audio Streams", *ISCA INTERSPEECH, (2018)*, Hyderabad, India, Sept. 2-6.

[CP5] H. Dubey, A. Sangwan and J. H. L. Hansen, "Robust Feature Clustering for Unsupervised Speech Activity Detection", *IEEE ICASSP, (2018)*, Calgary, Canada, pp. 2726–2730.

[**CP6**] **H. Dubey**, A. Sangwan and J. H. L. Hansen, "A Robust Diarization System for Measuring Dominance in Peer-Led Team Learning Groups", *IEEE SLT, (2016)*, San Diego, USA, pp. 319–323, *(Travel Grant Award).*

[CP7] H. Dubey, L. Kaushik, A. Sangwan and J. H. L. Hansen, "A Speaker Diarization System for Studying Peer-Led Team Learning Groups", *ISCA INTERSPEECH, (2016)*, San Francisco, USA, pp. 2180–2184.

List of Publications-II

[**Abs1**] J. H. L. Hansen, **H. Dubey**, A. Sangwan, L. Kaushik, V. Kothapally, "UTDallas-PLTL: Advancing multistream speech processing for interaction assessment in Peer-Led Team Learning sessions", *JASA, 2018*, 143 (3), 1869-1869.

[Abs2] J. H. L. Hansen, **H. Dubey**, and A. Sangwan, "CRSS-LDNN: Long-duration naturalistic noise corpus containing multi-layer noise recordings for robust speech processing", *JASA, (2018):* 144.3 1797-1797.

[Abs3] J. H. L. Hansen, A. Sangwan, A. Ziaei, **H. Dubey**, L. Kaushik and C. Yu, "Prof-Life-Log: Monitoring and assessment of human speech and acoustics using daily naturalistic audio streams", *JASA, (2016),* 140(4), 3010-3010,

[Abs4] J. H. L. Hansen, K. Hickman, N. Jones, **H. Dubey**, A. Sangwan, V. Kothapally, "UTDallas-PLTL: Leveraging Spoken Language Technology for Assessment of Communication based Learning Behavior in Peer-Led Team Learning", *Sixth Annual Conference Peer-Led Team Learning International Society's (PLTLIS), (2017)*, Northeastern Illinois University in Chicago, Illinois, USA, June 1-3.

REFERENCES

Ajmera, J., H. Bourlard, and I. Lapidot (2002). Improved unknown-multiple speaker clustering using hmm. Technical report, IDIAP.

Ajmera, J., I. McCowan, and H. Bourlard (2004). Robust speaker change detection. *IEEE Signal Processing Letters 11*(8), 649–651.

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

Akaike, H. (1981). Likelihood of a model and information criteria. *Journal of econometrics 16*(1), 3–14.

Alam, J., P. Kenny, P. Ouellet, T. Stafylakis, and P. Dumouchel (2014). Supervised/unsupervised voice activity detectors for text-dependent speaker recognition on the RSR2015 corpus. In *ISCA Odyssey Speaker and Language Recognition Workshop*, pp. 123–130.

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

Anh, N. K., N. T. Tam, and N. Van Linh (2013). Document clustering using mixture model of von Mises-Fisher distributions on document manifold. In *International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 140–145. IEEE.

Arons, B. Techniques, perception, and applications of time-compressed speech. In *Proceedings of 1992 Conference.*

Arons, B. (1994). Pitch-based emphasis detection for segmenting speech recordings. In *ICSLP.*

Arons, B. (1997). Speechskimmer: a system for interactively skimming recorded speech. *ACM Trans. on Computer-Human Interaction 4*(1), 3–38.

Ba, J. L., J. R. Kiros, and G. E. Hinton (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Banerjee, A., I. Dhillon, J. Ghosh, and S. Sra (2003). Generative model-based clustering of directional data. In *ACM International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pp. 19–28.

Banerjee, A., I. Dhillon, J. Ghosh, and S. Sra (2005). Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research 6*, 1345–1382.

Basu, S., T. Choudhury, B. Clarkson, A. Pentland, et al. (2001). Learning human interactions with the influence model. NIPS.

Bezdek, J. C. (2013). *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media.

Bozdogan, H. (2000). Akaike's information criterion and recent developments in information complexity. *Journal of mathematical psychology 44*(1), 62–91.

Camacho, A. and J. G. Harris (2008). A sawtooth waveform inspired pitch estimator for speech and music. *The Journal of the Acoustical Society of America 124*(3), 1638–1652.

Carletta, J., S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, et al. (2005). The ami meeting corpus: A pre-announcement. In *International Workshop on Machine Learning for Multimodal Interaction*, pp. 28–39. Springer.

Carlson, K., D. Turvold Celotta, E. Curran, M. Marcus, and M. Loe (2016). Assessing the Impact of a Multi-Disciplinary Peer-Led-Team Learning Program on Undergraduate STEM Education. *Journal of University Teaching & Learning Practice 13*(1), 5.

Castaldo, F., D. Colibro, E. Dalmasso, P. Laface, and C. Vair (2008). Stream-based speaker segmentation using speaker factors and eigenvoices. In *IEEE ICASSP*, pp. 4133–4136.

Chang, S.-Y. and N. Morgan (2014). Robust CNN-based speech recognition with Gabor filter kernels. In *ISCA INTERSPEECH*.

Chen, F. R. and M. Withgott (1992). The use of emphasis to automatically summarize a spoken discourse. In *IEEE ICASSP*.

Chi, T.-S. (2003). *Computational spectrotemporal auditory model with applications to acoustical information processing*. Ph. D. thesis, University of Maryland at College Park, USA.

Cournapeau, D., S. Watanabe, A. Nakamura, and T. Kawahara (2010). Online unsupervised classification with model comparison in the variational Bayes framework for voice activity detection. *IEEE Journal of Selected Topics in Signal Processing 4*(6), 1071–1083.

Cracolice, M. S. and J. C. Deming (2001). Peer-led team learning. *The Science Teacher 68*(1), 20.

Cummins, F. (2009). Rhythm as entrainment: The case of synchronous speech. *Journal of Phonetics 37*(1), 16–28.

Davis, S. and P. Mermelstein (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing 28*(4), 357–366.

De Cheveigné, A. and H. Kawahara (2002). YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America 111*(4), 1917–1930.

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

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

Dortet-Bernadet, J.-L. and N. Wicker (2007). Model-based clustering on the unit sphere with an illustration using gene expression profiles. *Biostatistics 9*(1), 66–80.

Drugman, T., Y. Stylianou, Y. Kida, and M. Akamine (2016). Voice activity detection: merging source and filter-based information. *IEEE Signal Processing Letters 23*(2), 252–256.

Dubey, H., L. Kaushik, A. Sangwan, and J. H. L. Hansen (2016). A Speaker Diarization System for Studying Peer-Led Team Learning Groups. In *ISCA INTERSPEECH*, pp. 2180–2184.

Dubey, H., A. Sangwan, and J. H. Hansen. Robust speaker clustering using mixtures of von mises-fisher distributions for naturalistic audio streams. In *ISCA INTERSPEECH*.

Dubey, H., A. Sangwan, and J. H. L. Hansen (2016a). A robust diarization system for measuring dominance in peer-led team learning groups. In *IEEE Spoken Language Technology Workshop (SLT)*, pp. 319–323.

Dubey, H., A. Sangwan, and J. H. L. Hansen (2016b). A robust diarization system for measuring dominance in peer-led team learning groups. In *IEEE Spoken Language Technology Workshop (SLT)*, pp. 319–323.

Dubey, H., A. Sangwan, and J. H. L. Hansen (2017). Using speech technology for quantifying behavioral characteristics in peer-led team learning sessions. *Computer Speech & Language 46*, 343–366.

Dubey, H., A. Sangwan, and J. H. L. Hansen (2018). Leveraging Frequency-Dependent Kernel and DIP-based Clustering for Robust Speech Activity Detection in Naturalistic Audio Streams. *IEEE/ACM Trans. on Audio, Speech and Language Processing*, –.

Dubey, H., A. Sangwan, and J. H. L. Hansen (2019). Transfer learning using raw waveform sincnet for robust speaker diarization. In *IEEE ICASSP, Brighton, UK.*

Dunbar, N. E. and J. K. Burgoon (2005). Perceptions of power and interactional dominance in interpersonal relationships. *Journal of Social and Personal Relationships 22*(2), 207–233.

Dupuy, G., S. Meignier, P. Deléglise, and Y. Estéve (2014). Recent Improvements on ILP-based Clustering for Broadcast News Speaker Diarization. In *ISCA Odyssey*, pp. 187–193.

Falk, T. H., C. Zheng, and W. Chan (2010). A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech. *IEEE Trans. on Audio, Speech, and Language Processing 18*(7), 1766–1774.

Ferrer, L., M. Graciarena, and V. Mitra (2016). A phonetically aware system for speech activity detection. In *IEEE ICASSP*.

Fraley, C. and A. E. Raftery (2006). Mclust version 3: an r package for normal mixture modeling and model-based clustering. Technical report, Washington Univ. Seattle Dept. of Statistics.

Fredouille, C. and G. Senay (2006). Technical improvements of the e-hmm based speaker diarization system for meeting records. In *MLMI*, Volume 4299, pp. 359–370. Springer.

Fukuda, T., O. Ichikawa, and M. Nishimura (2010). Long-term spectro-temporal and static harmonic features for voice activity detection. *IEEE Journal of Selected Topics in Signal Processing 4*(5), 834–844.

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

Garofolo, J. (1993). *DARPA TIMIT: acoustic-phonetic continuous speech corpus CD-ROM*. US Dept. of Commerce, National Institute of Standards and Technology.

Gebru, I. D., S. Ba, X. Li, and R. Horaud (2018). Audio-visual speaker diarization based on spatiotemporal bayesian fusion. *IEEE Trans. on pattern analysis and machine intelligence 40*(5), 1086–1099.

Gehring, J., Y. Miao, F. Metze, and A. Waibel (2013). Extracting deep bottleneck features using stacked auto-encoders. In *IEEE ICASSP*.

Ghosh, P., A. Tsiartas, and S. Narayanan (2011). Robust voice activity detection using long-term signal variability. *IEEE Trans. on Audio, Speech, and Language Processing 19*(3), 600–613.

Glorot, X. and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256.

Gonzalez, S. and M. Brookes (2014). PEFAC-a pitch estimation algorithm robust to high levels of noise. *IEEE Trans. on Audio, Speech, and Language Processing 22*(2), 518–530.

Górriz, J. M., J. Ramírez, E. W. Lang, and C. G. Puntonet (2006). Hard C-means clustering for voice activity detection. *Speech Communication 48*(12), 1638–1649.

Graciarena, M., A. Alwan, D. Ellis, H. Franco, L. Ferrer, J. H. L. Hansen, A. Janin, B. S. Lee, Y. Lei, V. Mitra, et al. (2013). All for one: feature combination for highly channel-degraded speech activity detection. In *ISCA INTERSPEECH*, pp. 709–713.

Graciarena, M., L. Ferrer, and V. Mitra (2016). The SRI system for the NIST OpenSAD 2015 speech activity detection evaluation. In *ISCA INTERSPEECH*, pp. 3673–3677.

Gupta, R., D. Bone, S. Lee, and S. Narayanan (2016). Analysis of engagement behavior in children during dyadic interactions using prosodic cues. *Computer Speech & Language 37*, 47–66.

Hallac, D., S. Vare, S. Boyd, and J. Leskovec (2017). Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 215–223. ACM.

Han, K. and D. Wang (2014). Neural network based pitch tracking in very noisy speech. *IEEE Trans. on Audio, Speech, and Language Processing 22*(12), 2158–2168.

Hansen, J. H., H. Dubey, and A. Sangwan (2018). Crss-ldnn: Long-duration naturalistic noise corpus containing multi-layer noise recordings for robust speech processing. *The Journal of the Acoustical Society of America 144*(3), 1797–1797.

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

Hansen, J. H., A. Sangwan, A. Ziaei, H. Dubey, L. Kaushik, and C. Yu (2016). Prof-Life-Log: Monitoring and assessment of human speech and acoustics using daily naturalistic audio streams. *The Journal of the Acoustical Society of America 140*(4), 3010–3010.

Hansen, J. H. L., H. Dubey, A. Sangwan, L. Kaushik, and V. Kothapally (2018). UTDallas-PLTL: Advancing multi-stream speech processing for interaction assessment in peer-led team learning. *The Journal of the Acoustical Society of America 143*(3), 1869–1869.

Hansen, J. H. L., K. Hickman, N. Jones, H. Dubey, A. Sangwan, and V. Kothapally (2017). UTDallas-PLTL: Leveraging Spoken Language Technology for Assessment of Communication based Learning Behavior in Peer-Led Team Learning. *Sixth Annual Conference Peer-Led Team Learning International Society, Northeastern Illinois University, Chicago, Illinois, USA.*, 5–10.

Harris, F. J. (1969). Windows, harmonic analysis, and the discrete fourier transform. *NUC TP532, Naval Undersea Center, San Diego, California, USA*.

Hartigan, J. A. and P. M. Hartigan (1985). The dip test of unimodality. *The Annals of Statistics*, 70–84.

Hartigan, P. M. (1985). Algorithm AS 217: Computation of the Dip Statistic to Test for Unimodality. *Journal of the Royal Statistical Society, Series C (Applied Statistics) 34* (3), 320–325.

Hermansky, H. and N. Morgan (1994). RASTA processing of speech. *IEEE Trans. on Audio, Speech, and Language Processing 2* (4), 578–589.

Hinton, G., O. Vinyals, and J. Dean (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hinton, G. E. and R. R. Salakhutdinov (2006). Reducing the dimensionality of data with neural networks. *Science 313* (5786), 504–507.

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

Huijbregts, M. and D. A. van Leeuwen (2012). Large-scale speaker diarization for long recordings and small collections. *IEEE Trans. on Audio, Speech, and Language Processing 20* (2), 404–413.

Hung, H., Y. Huang, G. Friedland, and D. Gatica-Perez (2008). Estimating the dominant person in multi-party conversations using speaker diarization strategies. In *IEEE ICASSP*.

Hung, H., D. B. Jayagopi, C. Yeo, G. Friedland, S. O. Ba, J.-M. Odobez, K. Ramchandran, N. Mirghafori, and D. Gatica-Perez (2007). Using audio and video features to classify the most dominant person in a group meeting. Number LIDIAP-CONF-2007-016.

Ioffe, S. and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jayagopi, D. B., H. Hung, C. Yeo, and D. Gatica-Perez (2009). Modeling dominance in group conversations using nonverbal activity cues. *IEEE Trans. on Audio, Speech, and Language Processing 17*(3), 501–513.

Jiao, Y., V. Berisha, M. Tu, T. T. Huston, and J. Liss (2015). Estimating speaking rate in spontaneous discourse. In *49th Asilomar Conference on Signals, Systems and Computers*.

Karakos, D., S. Novotney, L. Zhang, and R. Schwartz (2016). Model adaptation and active learning in the BBN speech activity detection system for the DARPA RATS program. In *ISCA INTERSPEECH*, pp. 3678–3682.

Kelly, F. and J. H. Hansen (2016). Evaluation and calibration of lombard effects in speaker verification. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 205–209. IEEE.

Kelly, F. and J. H. Hansen (2018). Detection and calibration of whisper for speaker recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1060–1065. IEEE.

Kennedy, L. S. and D. P. W. Ellis (2003). Pitch-based emphasis detection for characterization of meeting recordings. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

Kesemen, O., Ö. Tezel, and E. Özkul (2016). Fuzzy c-means clustering algorithm for directional data (fcm4dd). *Expert systems with applications 58*, 76–82.

Khoury, E. and M. Garland (2016). I-vectors for speech activity detection. In *ISCA Odyssey Speaker and Language Recognition Workshop*, pp. 334–339.

Kim, C. and R. M. Stern (2008). Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. In *ISCA INTERSPEECH*, pp. 2598–2601.

Kinnunen, T. and P. Rajan (2013). A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data. In *IEEE ICASSP*, pp. 7229–7233.

Kinnunen, T., A. Sholokhov, E. Khoury, D. Thomsen, M. Sahidullah, and Z.-H. Tan (2016). HAPPY team entry to NIST OpenSAD challenge: a fusion of short-term unsupervised and segment i-vector based speech activity detectors. In *ISCA INTERSPEECH*, pp. 2992–2996.

Kinoshita, K., M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, et al. (2016). A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research. *EURASIP Journal on Advances in Signal Processing 2016*(1), 1–19.

Koller, D., N. Friedman, and F. Bach (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kotti, M., V. Moschou, and C. Kotropoulos (2008). Speaker segmentation and clustering. *Signal processing 88*(5), 1091–1124.

Larrue, J. and A. Trognon (1993). Organization of turn-taking and mechanisms for turn-taking repairs in a chaired meeting. *Journal of Pragmatics 19*(2), 177–196.

Lauritzen, S. L. (1996). *Graphical models*, Volume 17. Clarendon Press.

Lee, B. S. and D. P. W. Ellis (2012). Noise robust pitch tracking by subband autocorrelation classification. In *ISCA INTERSPEECH*.

Lee, K. A., A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. v. Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma, et al. (2015). The RedDots data collection for speaker recognition. In *ISCA INTERSPEECH*, pp. 2996–3000.

Li, M., A. Tsiartas, M. Van Segbroeck, and S. S. Narayanan (2013). Speaker verification using simplified and supervised i-vector modeling. In *IEEE ICASSP*, pp. 7199–7203.

Li, Q. and Y. Huang (2011). An auditory-based feature extraction algorithm for robust speaker identification under mismatched conditions. *IEEE Trans. on Audio, Speech, and Language Processing 19*(6), 1791–1801.

Liu, W., Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song (2017). Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220.

Liu, Y., H. Li, and X. Wang (2017). Rethinking feature discrimination and polymerization for large-scale recognition. *arXiv preprint arXiv:1710.00870*.

Lyle, K. S. and W. R. Robinson (2003). A statistical evaluation: Peer-led team learning in an organic chemistry course. *Journal of Chemical Education 80*(2), 132.

Maaten, L. V. D. and G. Hinton (2008). Visualizing data using t-SNE. *Journal of machine learning research 9*(Nov), 2579–2605.

Madikeri, S. and H. Bourlard (2015). Kl-hmm based speaker diarization system for meetings. In *IEEE ICASSP)*.

Mardia, K. V., C. C. Taylor, and G. K. Subramaniam (2007). Protein bioinformatics and mixtures of bivariate von Mises distributions for angular data. *Biometrics 63*(2), 505–512.

Mast, M. S. (2002). Dominance as expressed and inferred through speaking time. *Human Communication Research 28*(3), 420–450.

Maurus, S. and C. Plant (2016). Skinny-dip: clustering in a sea of noise. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1055–1064.

McCowan, I., J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, et al. (2005). The AMI meeting corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, Volume 88, pp. 100.

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

Meignier, S., D. Moraru, C. Fredouille, J.-F. Bonastre, and L. Besacier (2006). Step-by-step and integrated approaches in broadcast news speaker diarization. *Computer Speech & Language 20*(2-3), 303–330.

Miao, Y. Pdnn: A python toolkit for deep learning. *URL https://www.cs.cmu.edu/ ymiao/pdnntk.html*.

Morgan, N. and E. Fosler-Lussier (1998). Combining multiple estimators of speaking rate. In *IEEE ICASSP*.

Nagy, P. and G. Németh (2016). Improving hmm speech synthesis of interrogative sentences by pitch track transformations. *Speech Communication 82*, 97–112.

Narayanan, S. and P. G. Georgiou (2013). Behavioral signal processing: Deriving human behavioral informatics from speech and language. *Proceedings of the IEEE 101*(5), 1203–1233.

Ng, T., B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselỳ, and P. Matejka (2012). Developing a speech activity detection system for the DARPA RATS program. In *ISCA INTERSPEECH*, pp. 1969–1972.

NIST NIST. NIST OpenSAD Challenge 2015. (Date last accessed 5-July-2016).

NIST NIST. NIST Pilot Speech Analytic Technologies Evaluation, OpenSAT 2017. (Date last accessed 25-Oct-2017).

NIST NIST. NIST STNR, Speech Signal to Noise Ratio, https://www.nist.gov/information-technology-laboratory/iad/mig/nist-speech-signal-noise-ratio-measurements.

NIST NIST. The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan.

Novotney, S., D. Karakos, J. Silovsky, and R. Schwartz (2016). BBN technologies' OpenSAD system. In *IEEE Spoken Language Technology Workshop*, pp. 8–12.

Nunes, J. A. C., D. Macêdo, and C. Zanchettin (2019). Additive margin sincnet for speaker recognition. *arXiv preprint arXiv:1901.10826*.

Pan, S. J. and Q. Yang (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering 22*(10), 1345–1359.

Panayotov, V., G. Chen, D. Povey, and S. Khudanpur (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. IEEE.

Parthasarathy, S. and C. Busso (2018). Ladder networks for emotion recognition: Using unsupervised auxiliary tasks to improve predictions of emotional attributes. *arXiv preprint arXiv:1804.10816*.

Patino, J., R. Yin, H. Delgado, H. Bredin, A. Komaty, G. Wisniewski, C. Barras, N. Evans, and S. Marcel (2018). Low-latency speaker spotting with online diarization and detection. In *ISCA Odyssey*, pp. 140–146.

Ramırez, J., J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio (2004). Efficient voice activity detection algorithms using long-term speech information. *Speech communication 42*(3), 271–287.

Ramírez, J., J. C. Segura, C. Benítez, L. García, and A. Rubio (2005). Statistical voice activity detection using a multiple observation likelihood ratio test. *IEEE Signal Processing Letters 12*(10), 689–692.

Ranjan, R., C. D. Castillo, and R. Chellappa (2017). L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*.

Ravanelli, M. and Y. Bengio (2018). Speaker recognition from raw waveform with sincnet. In *IEEE SLT*.

Renner, B. (2006). Curiosity about people: The development of a social curiosity measure in adults. *Journal of personality assessment 87*(3), 305–316.

Reynolds, D. A., T. F. Quatieri, and R. B. Dunn (2000). Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing 10*(1-3), 19–41.

Rienks, R., D. Zhang, D. Gatica-Perez, and W. Post (2006). Detection and application of influence rankings in small group meetings. In *8th international conference on Multimodal interfaces*. ACM.

Roh, Y. S., M. Kelly, and E. H. Ha (2016). Comparison of instructor-led versus peer-led debriefing in nursing students. *Nursing & health sciences*.

Rosenblatt, M. (1956). A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences 42*(1), 43–47.

Ryant, N., K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman (2018). First dihard challenge evaluation plan.

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

Sahami, M. and T. D. Heilman (2006). A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pp. 377–386. ACM.

Sahidullah, M. and G. Saha (2012). Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech Communication 54*(4), 543–565.

Sainath, T. N., R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals (2015). Learning the speech front-end with raw waveform CLDNNs. In *ISCA INTERSPEECH*.

Salmun, I., I. Shapiro, I. Opher, and I. Lapidot (2017). PLDA-based mean shift speakers' short segments clustering. *Computer Speech & Language 45*, 411–436.

Sangwan, A., J. H. L. Hansen, D. W. Irvin, S. Crutchfield, and C. R. Greenwood (2015). Studying the relationship between physical and language environments of children: Who's speaking to whom and where? In *IEEE Signal Proc.Education Workshop, Salt Lake City, Utah*, pp. 49–54.

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

Sarikaya, R. and J. H. L. Hansen (1998). Robust speech activity detection in the presence of noise. In *ICSLP Inter. Conf. On Spoken Language Processing*.

Schafer, R. W. (2011). What is a savitzky-golay filter? *IEEE Signal Processing Magazine 28*(4), 111–117.

Schluter, R., I. Bezrukov, H. Wagner, and H. Ney (2007). Gammatone features and feature combination for large vocabulary speech recognition. In *IEEE ICASSP*, pp. IV–649–IV–652.

Seki, H., K. Yamamoto, and S. Nakagawa (2017). A deep neural network integrated with filterbank learning for speech recognition. In *IEEE ICASSP*, pp. 5480–5484.

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

Senoussaoui, M., P. Kenny, T. Stafylakis, and P. Dumouchel (2014). A study of the cosine distance-based mean shift for telephone speech diarization. *IEEE Trans. on Audio, Speech and Language Processing 22*(1), 217–227.

Shao, Y., Z. Jin, D. Wang, and S. Srinivasan (2009). An auditory-based feature for robust speech recognition. In *IEEE ICASSP*, pp. 4625–4628.

Shao, Y. and D. Wang (2008). Robust speaker identification using auditory features and computational auditory scene analysis. In *IEEE ICASSP*, pp. 1589–1592.

Shin, J. W., J.-H. Chang, and N. S. Kim (2010). Voice activity detection based on statistical models and machine learning approaches. *Computer Speech & Language 24*(3), 515–530.

Sholokhov, A., M. Sahidullah, and T. Kinnunen (2018). Semi-supervised speech activity detection with an application to automatic speaker verification. *Computer Speech & Language 47*, 132–156.

Sinclair, M. and S. King (2013). Where are the challenges in speaker diarization? In *IEEE ICASSP*, pp. 7741–7745.

Snyder, D., D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur (2018). X-vectors: Robust DNN embeddings for speaker recognition. *IEEE ICASSP*.

Snyder, J. J., J. D. Sloane, R. D. Dunk, and J. R. Wiles (2016). Peer-led team learning helps minority students succeed. *PLoS Biol 14*(3), e1002398.

Snyder, J. J. and J. R. Wiles (2015). Peer led team learning in introductory biology: Effects on peer leader critical thinking skills. *PloS one 10*(1), e0115084.

Sohn, J., N. S. Kim, and W. Sung (1999). A statistical model-based voice activity detection. *IEEE Signal Processing Letters 6*(1), 1–3.

Sun, H., B. Ma, S. Z. K. Khine, and H. Li (2010). Speaker diarization system for RT07 and RT09 meeting room audio. In *IEEE ICASSP*, pp. 4982–4985.

Taghia, J., Z. Ma, and A. Leijon (2013). On von-Mises Fisher mixture model in text-independent speaker identification. In *ISCA INTERSPEECH*, pp. 2499–2503.

Tan, Z.-H. and B. Lindberg (2010). Low-complexity variable frame rate analysis for speech recognition and voice activity detection. *IEEE Journal of Selected Topics in Signal Processing 4*(5), 798–807.

Thomas, S., G. Saon, M. Van Segbroeck, and S. S. Narayanan (2015). Improvements to the IBM speech activity detection system for the DARPA RATS program. In *IEEE ICASSP*, pp. 4500–4504.

Tien, L. T., V. Roth, and J. Kampmeier (2002). Implementation of a peer-led team learning instructional approach in an undergraduate organic chemistry course. *Journal of research in science teaching 39*(7), 606–632.

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.

Trigeorgis, G., F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou (2016). Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *IEEE ICASSP*, pp. 5200–5204.

Van Den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu (2016). WaveNet: A generative model for raw audio. In *SSW*, pp. 125.

Van Segbroeck, M., R. Travadi, and S. S. Narayanan (2014). UBM fused total variability modeling for language identification. In *ISCA INTERSPEECH*, pp. 3027–3031.

Van Segbroeck, M., A. Tsiartas, and S. Narayanan. A robust frontend for vad: exploiting contextual, discriminative and spectral cues of human voice. In *ISCA INTERSPEECH*.

Varga, A. and H. J. Steeneken (1993). Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication 12*(3), 247–251.

Versteegh, M., R. Thiolliere, T. Schatz, X.-N. Cao, X. Anguera, A. Jansen, and E. Dupoux (2015). The zero resource speech challenge 2015. In *ISCA INTERSPEECH*, pp. 3169–3173.

Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol (2008). Extracting and composing robust features with denoising autoencoders. In *25th international conference on Machine learning*. ACM.

Walker, K. and S. Strassel (2012a). The RATS radio traffic collection system. In *ISCA Odyssey Speaker and Language Recognition Workshop*, pp. 291–297.

Walker, K. and S. Strassel (2012b). The RATS radio traffic collection system. In *ISCA Odyssey Speaker and Language Recognition Workshop*, pp. 291–297.

Wamser, C. C. (2006). Peer-led team learning in organic chemistry: Effects on student performance, success, and persistence in the course. *Journal of Chemical Education 83*(10), 1562.

Wang, D. and S. S. Narayanan (2007). Robust speech rate estimation for spontaneous speech. *IEEE Trans. on Audio, Speech, and Language Processing 15*(8), 2190–2201.

Wang, F., J. Cheng, W. Liu, and H. Liu (2018). Additive margin softmax for face verification. *IEEE Signal Processing Letters 25*(7), 926–930.

Wang, F., X. Xiang, J. Cheng, and A. L. Yuille (2017). Normface: l 2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049. ACM.

Wang, Y., H. Yao, and S. Zhao (2016). Auto-encoder based dimensionality reduction. *Neurocomputing 184*, 232–242.

Watson, G. N. (1995). *A treatise on the theory of Bessel functions*. Cambridge University Press.

Wen, Y., K. Zhang, Z. Li, and Y. Qiao (2016). A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pp. 499–515. Springer.

Wickerhauser, M. V. (1991). Lectures on wavelet packet algorithms. In *Lecture notes, INRIA*.

Wooters, C. and M. Huijbregts (2007). The icsi rt07s speaker diarization system. In *Multimodal Technologies for Perception of Humans*, pp. 509–519. Springer.

Wu, J. and X. Zhang (2011). Maximum margin clustering based statistical VAD with multiple observation compound feature. *IEEE Signal Processing Letters 18*(5), 283–286.

Yella, S. H. and A. Stolcke (2015). A comparison of neural network feature transforms for speaker diarization. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Yoshioka, T., T. Nakatani, M. Miyoshi, and H. G. Okuno (2011). Blind separation and dereverberation of speech mixtures by joint optimization. *IEEE Trans. on Audio, Speech, and Language Processing 19*(1), 69–84.

Young, K. (2016). *Handbook of social psychology*. Routledge.

Zeinali, H., H. Sameti, L. Burget, J. Cernockỳ, N. Maghsoodi, and P. Matejka (2016). i-Vector/HMM based text-dependent speaker verification system for RedDots challenge. In *ISCA INTERSPEECH*, pp. 440–444.

Zhang, C., K. Koishida, and J. H. Hansen (2018). Text-independent speaker verification based on triplet convolutional neural network embeddings. *IEEE TASLP 26*(9), 1633–1644.

Zhang, C., G. Liu, C. Yu, and J. H. Hansen (2015). I-vector based physical task stress detection with different fusion strategies. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Zhang, X.-L. and J. Wu (2013). Deep belief networks based voice activity detection. *IEEE Trans. on Audio, Speech, and Language Processing 21*(4), 697–710.

Zhong, S. (2005). Efficient online spherical k-means clustering. In *IEEE International Joint Conference on Neural Networks*, Volume 5, pp. 3180–3185.

Zhu, X., C. Barras, S. Meignier, and J. Gauvain (2005). Combining speaker identification and BIC for speaker diarization. In *ISCA INTERSPEECH*, pp. 2441–2444.

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

Ziaei, A., A. Sangwan, and J. H. L. Hansen (2016). Effective word count estimation for long duration daily naturalistic audio recordings. *Speech Communication 84*, 15–23.

BIOGRAPHICAL SKETCH

Harishchandra Dubey was born in Mirzapur, India on June 1, 1990. Harishchandra Dubey has been a graduate research assistant at the Robust Speech Technologies Lab (RSTL), Center for Robust Speech Systems (CRSS), at The University of Texas at Dallas (UT Dallas) since the Spring of 2016. Prior to joining the PhD program in Electrical Engineering at UT Dallas, during 2015 he was a Visiting Scholar in the department of Electrical, Computer and Biomedical Engineering at the University of Rhode Island (URI), Kingston, RI, USA affiliated with Wearable Biosensing Lab. He is a recipient of URI IP awards in 2016 and 2017 for his work on wearable IoT for voice and speech treatments of patients with Parkinson's disease. He was a research intern for The Siri speech team (Apple Inc.) in Cambridge, MA, USA during May-August 2017. He was a research intern for Audio and Acoustics research group at Microsoft Research, Redmond, WA, USA during May-August 2018. He received his Bachelor of Technology (B.Tech) degree in Electronics and Communication Engineering from Motilal Nehru National Institute of Technology (MNNIT) Allahabad, India and his Master of Science degree in Communication and Multimedia Engineering from Friedrich-Alexander University of Erlangen-Nuremberg (FAU), Germany where he was associated with the Audio and Acoustic Signal Processing group. He was awarded the STIBET Stipend, German Academic Exchange Service (DAAD) for his master's thesis on Robust Speech Recognition. Previously, he worked in several RandD teams including Siemens Ltd. India, Fraunhofer IIS Erlangen, International AudioLabs, Erlangen, and Siemens Healthcare Erlangen (department of Angiography and Interventional X-Ray Systems, and Imaging Solutions), all in Germany. His current research interests include Speech Activity Detection, Speaker Diarization, Recognition and Verification, Speech Recognition, and Behavioral Informatics. His broader interests lies in Machine Learning for Audio and Speech Processing, Biomedical Signal Processing, Voice and Speech Therapy, Wearable Systems, Fog computing, and Internet-of-Things (IoT).

CURRICULUM VITAE

Harishchandra Dubey

May 23, 2019

Contact Information:

Center for Robust Speech Systems
Department of Electrical Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: harish.dubey123@gmail.com
harishchandra.dubey@fau.de
harishchandra.dubey@utdallas.edu

Educational History:

B.Tech., Electronics and Communication Engineering, Motilal Nehru National Institute of Technology (MNNIT) Allahabad, India, 2012
M.Sc., Communication and Multimedia Engineering, Friedrich Alexander University of Erlangen-Nuremberg (FAU), Germany, 2015
Ph.D., Electrical Engineering, Center for Robust Speech Systems (CRSS), The University of Texas at Dallas, USA, 2019

Ph.D. Electrical Engineering Dissertation titled *Deep Neural Networks and Model-based Approaches for Robust Speaker Diarization in Naturalistic Audio Streams*, The University of Texas at Dallas
Advisors: Prof. John H. L. Hansen, Dr. Abhijeet Sangwan, Prof. Carlos Busso, Dr. P. K. Rajasekaran, Dr. Chin-Tuan Tan

Employment History:

Machine Learning Scientist-II, Microsoft Corporation, Redmond, Washington, WA, USA. May 2019 – present

Graduate Research Assistant, Center for Robust Speech Systems [Prof. John H.L. Hansen], The University of Texas at Dallas, USA. January 2016 – May 2019

Research Intern, Audio and Acoustics Research Group, Microsoft Research, Redmond, WA, USA. May –Aug. 2018

Research Intern, Apple-Siri, Cambridge, Massachusetts, USA. May –Aug. 2017

Visiting Scholar, Wearable Biosensing Lab, Dept. of Electrical, Computer and Biomedical Engineering, University of Rhode Island, USA. April –Dec. 2015

Research Assistant, SIEMENS Healthcare Imaging and Therapy Systems (Angiography and Interventional X-Ray Systems), Erlangen, Germany. Dec. 2013 –April 2015

Research Assistant (Spatial Audio Signal Processing), Prof. Dr. ir. Emanuel Habets, International Audio Laboratories, Erlangen, Germany. June 2014 –Jan. 2015

Research Assistant (Video Group), Fraunhofer IIS, Erlangen, Germany. Dec.2013 –May 2014

Graduate Engineer, Siemens Ltd. India. July 2012 –Aug. 2013

Research Intern, CRRAO Advanced Institute of Mathematics, Statistics and Computer Science, Hyderabad, India. May –July 2011

Professional Recognition and Honors:

IEEE Signal Processing Society (SPS) Travel Grant for attending IEEE SLT workshop, Dec. 2016
Intellectual Property Award, University of Rhode Island, USA, Feb. 2017
Intellectual Property Award, University of Rhode Island, USA, Feb. 2016
STIBET Stipend, German Academic Exchange Service (DAAD) for Master Thesis (http://www.daad.de/), Oct. 2014

Professional Memberships:

Institute of Electrical and Electronics Engineers (IEEE), 2015–present
International Speech Communication Association (ISCA), 2016–present
Society for Industrial and Applied Mathematics (SIAM), 2018–present
Member of International Speech Communication Association Student Advisory Committee (ISCA-SAC) http://www.isca-students.org, 2018–present