CASA-BASED ROBUST SPEAKER IDENTIFICATION

DISSertation

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By
Xiaojia Zhao, M.S.
Graduate Program in Computer Science and Engineering

The Ohio State University
2014

Dissertation Committee:
Professor DeLiang Wang, Advisor
Professor Eric Fosler-Lussier
Professor Mikhail Belkin
ABSTRACT

As a primary topic in speaker recognition, speaker identification (SID) aims to identify the underlying speaker(s) given a speech utterance. SID systems perform well under matched training and test conditions. In real-world environments, mismatch caused by background noise, room reverberation or competing voice significantly degrades the performance of such systems. Achieving robustness to the SID systems becomes an important research problem. Existing approaches address this problem from different perspectives such as proposing robust speaker features, introducing noise to clean speaker models, and using speech enhancement methods to restore clean speech characteristics. Inspired by auditory perception, computational auditory scene analysis (CASA) typically segregates speech from interference by producing a time-frequency mask. This dissertation aims to address the SID robustness problem in the CASA framework.

We first deal with the noise robustness of SID systems. We employ an auditory feature, gammatone frequency cepstral coefficient (GFCC), and show that this feature captures speaker characteristics and performs substantially better than conventional speaker features under noisy conditions. To deal with noisy speech, we apply CASA separation and then either reconstruct or marginalize corrupted components indicated by a CASA mask. We find that both reconstruction and marginalization are effective. We
further combine these two methods into a single system based on their complementary advantages, and this system achieves significant performance improvements over related systems under a wide range of signal-to-noise ratios (SNR). In addition, we conduct a systematic investigation on why GFCC shows superior noise robustness and conclude that nonlinear log rectification is likely the reason.

Speech is often corrupted by both noise and reverberation. There have been studies to address each of them, but the combined effects of noise and reverberation have been rarely studied. We address this issue in two phases. We first remove background noise through binary masking using a deep neural network (DNN) classifier. Then we perform robust SID with speaker models trained in selected reverberant conditions, on the basis of bounded marginalization and direct masking. Evaluation results show that the proposed method substantially improves SID performance compared to related systems in a wide range of reverberation time and SNRs.

The aforementioned studies handle mixtures of target speech and non-speech intrusions by taking advantage of their different characteristics. Such methods may not apply if the intrusion is a competing voice, which is of similar characteristics as the target. SID in cochannel speech, where two speakers are talking simultaneously over a single recording channel, is a well-known challenge. Previous studies address this problem in the anechoic environment under the Gaussian mixture model (GMM) framework. On the other hand, cochannel SID in reverberant conditions has not been addressed. This dissertation studies cochannel SID in both anechoic and reverberant conditions. We first investigate GMM-based approaches and propose a combined system that integrates two cochannel SID methods. Secondly, we explore DNNs for cochannel
SID and propose a DNN-based recognition system. Evaluation results demonstrate that our proposed systems significantly improve SID performance over recent approaches in both anechoic and reverberant conditions and various target-to-interferer ratios.
Dedicated to my mother, Qiuyue Qu, my father, Hongqi Zhao and my wife, Na Zhang
ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my advisor, Professor DeLiang Wang. Without his support and guidance, I would not be here finishing up my Ph.D. degree. He leads by example. His integrity and expertise have always been inspirational to me in research and life. His instructions help me make strides in professional writing, although I still have a long way to go.

I would like to sincerely thank Professor Eric Fosler-Lussier and Professor Mikhail Belkin. Professor Fosler-Lussier has taught me the fundamentals of automatic speech recognition and artificial intelligence. His insightful questions and comments during my candidacy examination make me think deeper in my subsequent research. I have great pleasure of attending Professor Belkin’s machine learning class and enriching my knowledge in machine learning. It is always a good learning experience by listening to his talks and discussions during AI seminars. Furthermore, I am grateful for his referral of me to Professor Wang in 2008. It is this referral and Professor Wang’s acceptance that help me come to Ohio State and enjoy this remarkable six year journey.

I want to thank my PNL labmates. Dr. Zhaozhang Jin, Dr. John Woodruff and Dr. Ke Hu offered me enormous help in my first few years. I have learned a lot from Dr. Yang Shao and we jointly wrote my first academic paper, which is a milestone in my Ph.D.
career. My fellow Ph.D. candidates, Arun Narayanan and Kun Han, came to the lab in the same year as me. They are the examples I look up to in research. Working with them helps me improve my research skills. Many thanks are due to Yuxuan Wang, my junior labmate. His expertise on machine learning, especially deep learning, helps shape my research directions. I have the pleasure to co-author a few papers with him, which are described in this dissertation.

My heartfelt acknowledgement must go to my parents and my wife, Na. Their endless support propels me to pursue my dream no matter what happens. They are and will always be the highest priority of my life.
VITA

April, 1988 .................................................. Born in Henan, China

June, 2008 .................................................. B.E. in Software Engineering,  
Nankai University,  
Tianjin, China

June, 2012 .................................................. M.S., Computer Science and Engineering,  
The Ohio State University

PUBLICATIONS

Journal Articles

X. Zhao, Y. Wang and D.L. Wang, “Cochannel speaker identification in anechoic and reverberant conditions”, in preparation


Conference Papers


**FIELDS OF STUDY**

**Major Field:** Computer Science and Engineering
# TABLE OF CONTENTS

ABSTRACT ......................................................................................................................... ii

ACKNOWLEDGMENTS .................................................................................................... vi

VITA .................................................................................................................................. viii

LIST OF TABLES ............................................................................................................. xiii

LIST OF FIGURES ........................................................................................................... xvi

CHAPTER 1 INTRODUCTION ......................................................................................... 1
  1.1 Motivation ................................................................................................................... 1
  1.2 Objectives .................................................................................................................. 6
  1.3 Organization of Dissertation ...................................................................................... 8

CHAPTER 2 BACKGROUND ......................................................................................... 10
  2.1 Basics of Automatic Speaker Recognition .............................................................. 10
  2.2 Noise-robust Speaker Recognition .......................................................................... 13
  2.3 Speaker Recognition in Noisy and Reverberant Conditions .................................... 15
  2.4 Cochannel Speaker Recognition .............................................................................. 18
  2.5 Speaker Recognition with Channel Distortions ....................................................... 21
  2.6 Summary .................................................................................................................. 23

CHAPTER 3 NOISE-ROBUST SPEAKER IDENTIFICATION ........................................ 24
  3.1 Introduction ............................................................................................................... 24
  3.2 System Overview ..................................................................................................... 25
  3.3 Feature Extraction and Mask Estimation .................................................................. 27
    3.3.1 Auditory Features .............................................................................................. 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>CASA-based Mask Estimation</td>
<td>29</td>
</tr>
<tr>
<td>3.4</td>
<td>Reconstruction Module</td>
<td>32</td>
</tr>
<tr>
<td>3.5</td>
<td>Marginalization Module</td>
<td>34</td>
</tr>
<tr>
<td>3.6</td>
<td>Combined System</td>
<td>36</td>
</tr>
<tr>
<td>3.7</td>
<td>Evaluation and Comparison</td>
<td>37</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Experiment Setup</td>
<td>38</td>
</tr>
<tr>
<td>3.7.2</td>
<td>GFCC Dimensions and Dynamic Features</td>
<td>39</td>
</tr>
<tr>
<td>3.7.3</td>
<td>Baseline Comparisons</td>
<td>41</td>
</tr>
<tr>
<td>3.7.4</td>
<td>Evaluation Results</td>
<td>43</td>
</tr>
<tr>
<td>3.7.5</td>
<td>Comparison with a Related System</td>
<td>47</td>
</tr>
<tr>
<td>3.8</td>
<td>Analyzing Noise Robustness of MFCC and GFCC Features</td>
<td>50</td>
</tr>
<tr>
<td>3.8.1</td>
<td>Differences in MFCC and GFCC Derivations</td>
<td>50</td>
</tr>
<tr>
<td>3.8.2</td>
<td>Diagnosis and Experimental Results</td>
<td>52</td>
</tr>
<tr>
<td>3.8.3</td>
<td>Further Exploration</td>
<td>59</td>
</tr>
<tr>
<td>3.9</td>
<td>Discussion</td>
<td>62</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>65</td>
</tr>
<tr>
<td>4.2</td>
<td>System Overview and Front-end Processing</td>
<td>66</td>
</tr>
<tr>
<td>4.2.1</td>
<td>System Overview</td>
<td>66</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Mask Definitions in Noisy and Reverberant Conditions</td>
<td>67</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Mask Estimation via DNN</td>
<td>68</td>
</tr>
<tr>
<td>4.3</td>
<td>Recognition Methodology and System Design</td>
<td>70</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Bounded Marginalization Module</td>
<td>70</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Direct Masking Module</td>
<td>71</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Reverberant Model Training</td>
<td>72</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Multi-condition Fusion and Module Combination</td>
<td>72</td>
</tr>
<tr>
<td>4.4</td>
<td>Evaluation and Comparison</td>
<td>74</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Experimental Setup</td>
<td>74</td>
</tr>
<tr>
<td>4.4.2</td>
<td>IBM Comparisons</td>
<td>78</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Performance with Estimated IBM</td>
<td>82</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Comparison with Related Systems</td>
<td>86</td>
</tr>
<tr>
<td>4.4.5</td>
<td>Evaluation with Real Impulse Responses</td>
<td>90</td>
</tr>
</tbody>
</table>
4.5 Discussion .............................................................................................................. 93

CHAPTER 5 COCHANNEL SPEAKER IDENTIFICATION IN ANECHOIC AND
REVERBERANT CONDITIONS .............................................................................. 97

5.1 Introduction ........................................................................................................... 97
5.2 System Overview .................................................................................................. 98
5.3 Problem Formulation and Identification Methodology ............................................ 100
  5.3.1 Problem Formulation ....................................................................................... 100
  5.3.2 Combined Method .......................................................................................... 104
  5.3.3 DNN-based Cochannel SID .......................................................................... 106
  5.3.4 Model Training ............................................................................................... 107
5.4 Evaluation and Comparison ................................................................................... 108
  5.4.1 Experimental Setup ....................................................................................... 108
  5.4.2 Frame Selection for JSS and Max vs. Sum ....................................................... 110
  5.4.3 Performance on SSC Corpus .......................................................................... 112
  5.4.4 Performance on NIST SRE Dataset with 50 Speakers ................................... 113
  5.4.5 Performance on NIST SRE Dataset with 100 Speakers ................................ 115
  5.4.6 Scalability Study of GMM-based and DNN-based Approaches on
       Cochannel SID ................................................................................................. 116
5.5 Discussion ............................................................................................................. 118

CHAPTER 6 CONTRIBUTIONS AND FUTURE WORK ............................................. 120

6.1 Contributions ....................................................................................................... 120
6.2 Insights .................................................................................................................. 122
6.3 Future Study ......................................................................................................... 124

BIBLIOGRAPHY ....................................................................................................... 126
LIST OF TABLES

Table 3.1: SID accuracy (%) of the proposed methods. GFCC_REC denotes the
reconstruction module. And GF_MAR the marginalization module. ............ 44

Table 3.2: SID accuracy (%) of the proposed methods with the IBM....................... 45

Table 3.3: SID accuracy (%) of the combined system and baselines. Performance is
averaged across different SNR conditions. ........................................... 46

Table 3.4: Differences between MFCC and GFCC........................................... 52

Table 3.5: Benchmark performance of GFCC and MFCC in terms of speaker recognition
rates (%). ............................................................................................. 53

Table 3.6: Impact of pre-emphasis on GFCC and MFCC in terms of speaker recognition
rates (%). ............................................................................................. 54

Table 3.7: Impact of number of frequency bands/channels on GFCC and MFCC in terms
of speaker recognition rates (%). ......................................................... 55

Table 3.8: Impact of cepstral liftering on GFCC and MFCC in terms of speaker
recognition rates (%). .......................................................................... 56

Table 3.9: Impact of frequency scale on GFCC and MFCC in terms of speaker
recognition rates (%). .......................................................................... 57
Table 3.10: Impact of nonlinear rectification on GFCC and MFCC in terms of speaker recognition rates (%). ................................................................. 58

Table 3.11: Impact of T-F representation on GFCC and MFCC in terms of speaker recognition rates (%). ........................................................................ 60

Table 4.1: Sizes of the simulated rectangular rooms ........................................ 76

Table 4.2: Optimal LCs (dB) for different IBM definitions .............................. 80

Table 4.3: SID accuracy (%) of the three IBM definitions. Performance is averaged cross all the SNRs ........................................................................... 82

Table 4.4: Benchmark SID performance (%) of reverberant speaker models in the reverberant test set. Note that the first column is the same as the last column of Table 4.4 as both use anechoic speaker models to recognize reverberant test data ........................................................................................................ 83

Table 4.5: Benchmark SID performance (%) of anechoic speaker models. Noise is excluded in the test set ........................................................................ 83

Table 4.6: SID accuracy (%) of the proposed system using IBM_R. MFCC_DM denotes the direct masking module with MFCC features. GFCC_DM denotes the direct masking module with GFCC features. GF_BM denotes the bounded marginalization module. Comb. Syst. denotes the proposed system ........... 84

Table 4.7: SID performance (%) summary of the proposed system with different IBM definitions ........................................................................................................ 85

Table 5.1: Illustration of the two proposed combination methods ....................... 106

Table 5.2: SID accuracy (%) of JSS in different types of frames. Max operation is used to approximate the joint distribution except for the last row (sum operation). . 111
Table 5.3: SID accuracy (%) on SSC Corpus................................................................. 113
Table 5.4: SID accuracy (%) on NIST SRE dataset with 50 speakers ......................... 114
Table 5.5: SID accuracy (%) on reverberant NIST SRE dataset with 50 speakers ....... 115
Table 5.6: SID accuracy (%) on reverberant NIST SRE dataset with 100 speakers......116
LIST OF FIGURES

Figure 3.1: Schematic diagram of a CASA-based robust speaker identification system. 26

Figure 3.2: Illustrations of a cochleagram (top) and a spectrogram (bottom) of a clean speech utterance. Note the asymmetric frequency resolution at low and high frequencies in the cochleagram. .................................................. 28

Figure 3.3: Illustration of estimated IBM produced by a voiced speech segregation system. The top plot shows the cochleagram of an utterance mixed with white noise at 0 dB SNR. The bottom plot presents an estimated IBM from the mixture in the top plot, where 1 is shown as white and 0 as black. ............... 31

Figure 3.4: Illustrations of energy compaction by GFCCs. Plot (a) shows a cochleagram of an utterance. Plot (b) shows a GF frame at 1 sec of (a). The original GF is plotted as the solid line and the resynthesized GF by 23 GFCCs is plotted as the dashed line. Plot (c) presents the resynthesized cochleagram from (a) using 23 GFCCs. ......................................................... 40

Figure 3.5: SID performance of different baseline systems for 3 noises. ................. 42

Figure 3.6: SID accuracy (%) comparisons of the proposed marginalization module and Pulella et al.’s system. Both systems utilize the ideal binary mask for separation ................................................................. 48
Figure 3.7: SID accuracy (%) comparisons of the proposed combined system and Pullella
et al.’s system using estimated binary masks.................................................. 49

Figure 4.1: Schematic diagram of the proposed speaker identification system............... 67

Figure 4.2: SID accuracy (%) using IBMDS with different LCs. BM denotes bounded
marginalization, DM direct masking. (a) BM performance. (b) DM
performance. Each point in the figure is averaged across all the test SNRs.
Anechoic denotes speaker models trained in the anechoic condition. 300 ms,
600 ms and 900 ms denote speaker models trained in the corresponding T60.
Multi-condition Fusion combines the local decisions from the four sets of
speaker models. ............................................................................................... 79

Figure 4.3: Illustrations of 3 IBM definitions on a TIMIT sentence with 0 dB SNR and
500 ms T60. (a) IBM R with two LCs, 0 dB on the left and −4 dB on the right.
(b) IBM ER with two LCs, 0 dB on the left and −4 dB on the right. (c) IBMDS
with two LCs, 0 dB on the left and −12 dB on the right. 1 is shown as white
and 0 as black. .......................................................................................... 81

Figure 4.4: SID comparisons of the proposed system with related systems under factory
noise, destroyer engine room noise and speech shape noise using simulated
RIRs.................................................................................................................. 88

Figure 4.5: Comparison of IBM R and estimated IBM R of an utterance mixed with SSN
and real RIR (T60 = 300 ms) and SNR = 6 dB. .............................................. 91

Figure 4.6: SID comparisons of the proposed system with related systems under factory
noise, destroyer engine room noise and speech shape noise using real RIRs. 92
Figure 5.1: Schematic diagrams of two proposed cochannel speaker identification systems. (a). GMM-based system. (b). DNN-based system. .................................. 99

Figure 5.2: Scalability study of DNN and GMM-based approaches with respect to the number of speakers.......................................................... 117
CHAPTER 1
INTRODUCTION

1.1 Motivation

At a young age, I realized I had the ability of immediately telling which acquaintance was speaking on the other end of the telephone. My recognition was based on their distinctive voices rather than speech contents. In the entertainment industry, there are impersonators that specialize at mimicking the voices of celebrities to entertain the audience. In both cases, people’s voices are directly associated with their identities. They are known as voiceprint, reminiscent of the concept of fingerprint. Recognizing persons from their voices is called speaker recognition.

Speaker recognition is a natural gift of human listeners. Usually the content of speech is not important. The distinctiveness of human voice is due to a variety of reasons. For example, the physiological differences of speech production organs like larynx and vocal tract play a major role. In addition, the style of speaking like intonation and distinctive words/phrases also contributes to speaker recognition.

Many real-world applications benefit from speaker recognition. For instance, verification procedures are enforced for a user to gain access to a building or telephone-
based customer service systems. Instead of entering passwords or answering a number of security questions, one can use his/her voice as a key and a built-in speaker recognition system automatically identifies the speaker or verifies if the speaker is the claimed one. Another example is in forensic investigations, where speaker recognition can be applied to an audio signal to identify the persons of interest in the recording. Nowadays the majority of the smart devices start to build voice control features. Speaker recognition can be utilized to ensure that voice commands are generated by the owner of the device instead someone else.

Like automatic speech recognition, automatic speaker recognition has been widely studied over the past few decades. A speaker recognition system, performing either speaker identification (SID) or speaker verification (SV) tasks, normally comprises three processes: feature extraction, speaker modeling, and decision making (Campbell, 1997; Furui, 2009). Typically, extracted speaker features are short-time spectral/cepstral features such as short-time Fourier transform spectral features and mel-frequency cepstral coefficients (MFCC), or long-term features like prosodic features (Shriberg et al., 2005; Shriberg, 2007). Short-time features aim to capture vocal tract information while long-term features mainly extract the different speaking styles. As for speaker modeling, Gaussian mixture models (GMM) are commonly used to model speaker feature distributions (Reynolds, 1995), while support vector machines (SVMs) have been applied to speaker verification (Campbell et al., 2006). For SID, recognition decisions are usually made based on likelihoods of observing data given candidate speaker models. The decision process of SV typically compares the score of the claimed speaker with a threshold to either accept or reject the claimed speaker.
Automatic speaker recognition systems are able to achieve a high level of performance in a well-matched condition. However, the performance drops significantly as speech is distorted by interference (Gong, 2002; Shao and Wang, 2006a). The task of improving the robustness of such systems is known as robust speaker recognition. Speech enhancement methods and robust speaker features have been explored to achieve noise-robustness (Shao et al., 2007; Pulella et al., 2008; Wang et al., 2011; May et al., 2012b). An alternative approach seeks to improve robustness by modeling noise and combining it with clean speaker models (Matsui et al., 1996; Wong and Russell, 2001). Meanwhile, one can directly train speaker models in the noisy environment (Ming et al., 2007). Aside from additive noise, channel distortion is a commonly studied problem, especially for the SV task. State-of-the-art SV systems typically employ i-vector or joint factor analysis (JFA) on supervectors, which are high-dimensional feature vectors, to explicitly model both channel and speaker characteristics (Kenny, 2006; Kenny et al., 2007; Dehak et al., 2011). Other approaches seek to directly compensate for channel variation on supervectors (Kinnunen and Li, 2010).

In real-world acoustic environments, the speech arriving at our ears comprises not only direct sound but also its reflections from the surfaces such as walls, floors and ceilings. This is known as room reverberation. Basically the reverberant sound is a mixture of delayed and attenuated versions of the original direct sound. Reverberation is typically modeled as a convolution between the direct sound and a room impulse response (RIR). Many factors such as the geometric shape of the room and locations of sound sources and receivers jointly determine an RIR, which can be divided into three parts: direct sound, early reflections and late reflections. Early reflections correspond to
the first few bounces from the surfaces and late reflections characterize the longer delayed and attenuated bounces of the original sound. Early reflections cause spectral distortions called coloration and the late reflections smear the speech spectrum across time (Wu and Wang, 2006). Together they corrupt harmonic structure and formants of speech, and present a considerable challenge to speaker recognition systems. Speaker features such as modulation spectral features (Falk and Chan, 2010) and features that incorporate phase information (Wang and Nakagawa, 2009) have shown robustness against reverberation. Blind dereverberation algorithms have been used to restore the anechoic signal or the early part of reverberant speech (Sadjadi and Hansen, 2014). Alternatively, one can introduce reverberation to speaker models to reduce the mismatch caused by reverberation (Akula et al., 2009). Such systems are able to achieve reasonable performance in noise-free reverberant conditions, but they are not designed to simultaneously deal with additive noise (particularly non-stationary noise).

Another challenge to speaker recognition systems in real environments is a competing voice. In such cases, the interference exhibits similar statistical characteristics as the target speech, which makes it difficult to separate. Cochannel speech refers to two voices recorded simultaneously in a single communication channel. Determining the identities of speakers in the cochannel speech is referred to as the cochannel speaker recognition problem. Lovekin et al. (2001) proposed to identify usable speech that comes from the target speaker and then perform SID. Shao and Wang (2003, 2006a) identify usable speech using a multi-pitch tracking algorithm. Then they jointly search for speaker candidates and assignments of usable speech segments. Their system not only outputs speaker identities, but also groups segments into two streams, archiving cochannel speech
separation. Hershey et al. (2010) and Li et al. (2010) both developed a two-stage algorithm for cochannel SID. State-of-the-art performance was reported on the speech separation and recognition challenge (SSC) corpus, which was tailored for robust speech recognition (Cooke and Lee, 2006). It is unclear how these methods perform on a standard speaker recognition dataset. To our knowledge, there is no study on cochannel SID in reverberant conditions.

Human listeners perform robustly in adverse acoustic environments. The human ability to function well in adverse acoustic environments is due to a perceptual process termed auditory scene analysis (ASA). Inspired by ASA research, computational auditory scene analysis (CASA) aims to organize sound based on ASA principles. The superior performance of the auditory system motivates us to explore robust speaker recognition, in particular SID, from the perspective of CASA.


1.2 Objectives

This dissertation aims to develop SID systems in the presence of distortions. In the following, we lay out specific objectives based on the type of distortion discussed in the last section.

*Noise-robust SID.* Speaker models trained in clean conditions show a mismatch problem in noisy test conditions. Speech enhancement methods such as spectral subtraction do not handle non-stationary noises well. We explore CASA for noise-robust SID. CASA typically separates speech from noise by producing a binary time-frequency (T-F) mask. Our first objective is to effectively estimate T-F masks under a number of noise types and signal-to-noise ratios (SNRs). Given an estimated binary mask, how to incorporate it into a standard GMM-based SID system needs to be investigated. The second objective is to study the usage of CASA masks in SID systems. Shao and Wang (2007) proposed a novel speaker feature, gammatone frequency cepstral coefficients (GFCC). The third objective is to examine the noise-robustness of GFCC and make a comparison with other speaker features such as MFCC.

*Robust SID in noisy and reverberant conditions.* Robustness of SID systems to noise or reverberation alone has been studied, but not their combined effects. The first objective is to study how the combined effects of noise and reverberation affect SID performance. CASA masks in the noisy and reverberation conditions have different definitions. This leads to the second objective of investigating their respective utilities in robust SID. Reverberation is typically simulated using the image method (Allen and Berkley, 1979). How does the performance achieved in simulated reverberant
environments generalize to real reverberant conditions? The last objective seeks to test the proposed system in real reverberant environments.

*Cochannel SID in anechoic and reverberant conditions.* Recent cochannel SID systems achieve nearly perfect performance on the SSC corpus. It remains to be seen how they perform on a standard speaker recognition corpus. Therefore our first objective is to evaluate these systems in a speaker recognition corpus. Shao and Wang (2006a) proposed a cochannel SID method that combines identification and separation. Our second objective is to make a performance comparison of this method and other state-of-the-art systems, and explore whether a combined system can provide further improvement. Existing cochannel SID methods are all GMM-based. Deep neural networks (DNNs), a recently popular supervised learning machine, could be useful for cochannel SID. Our third objective is to make the first attempt of building a DNN-based cochannel SID system and compare its performance with GMM-based ones. As there is no study on cochannel SID in reverberant conditions, our next objective is to address this gap. In addition, the scalability of the proposed systems to speaker set size is critical for real applications, so our last objective is to systematically investigate the scalability of GMM-based and DNN-based approaches.
1.3 Organization of Dissertation

Thus far we have discussed the motivation and objectives of this dissertation. Chapter 2 presents the background of automatic speaker recognition, particularly robust speaker recognition. First we give an overview of standard speaker recognition technologies including the predominant GMM framework. Then we visit existing robust speaker recognition approaches categorized by their objectives.

Chapter 3 introduces a CASA-based noise robust SID system. The goal is to perform SID in various noisy environments. Auditory features, gammatone features (GF) and GFCC, are extracted using a front-end of gammatone filterbank. Noise separation is achieved through CASA masks estimated using multi-layer perceptrons (MLP). Two missing feature techniques, namely bounded marginalization and feature reconstruction, are investigated. We propose to combine these two to leverage their complementary advantages.

Chapter 4 proposes a system that deals with the combined effects of noise and reverberation. DNN-based mask estimation is explored for noise separation. We simulate multiple reverberant training conditions. Speaker models trained in these conditions are exploited collectively to handle unknown test reverberation. We employ bounded marginalization and direct masking to perform SID. Their combination is further investigated. We compare the proposed system with related systems in various testing conditions.

Chapter 5 addresses the cochannel SID problem. We first review existing GMM-based cochannel SID methods including systems that produce state-of-the-art
performance on the SSC corpus. We study two methods. One jointly performs cochannel SID and separation. The other achieves cochannel SID in two stages. We combine them for further performance improvement. Alternatively, we develop a DNN-based cochannel SID system by formulating cochannel SID as a multi-class classification problem. The proposed systems are compared with state-of-the-art methods on a standard speaker recognition corpus.

Chapter 6 discusses the insights gained from the dissertation, summarizes my contributions, and makes suggestions on future research directions.
CHAPTER 2
BACKGROUND

In this chapter, we first discuss the fundamentals of automatic speaker recognition. Then we review existing approaches concerned with robustness of speaker recognition.

2.1 Basics of Automatic Speaker Recognition

Speakers sound differently due to physiological differences such as vocal tract size, larynx size and other voice production organs, and speaking style differences such as accent, prosody and frequently used words. The task of automatic speaker recognition is to identify the underlying speaker or verify the claimed speaker from a sound recording, by exploiting these differences. Identifying the underlying speaker is known as SID. Verifying if a claimed speaker is indeed the underlying speaker is SV. In addition, speaker recognition is categorized into text-dependent and text-independent recognition based on whether to assume the knowledge of written text.

Speaker features encode speaker specific characteristics, and are extracted from time-domain signals. Commonly used speaker features include short-time spectral/cepstral features, spectro-temporal features, prosodic features, etc. Short-time features are
generally derived from short-time Fourier transform (STFT). Specifically, time domain signals are broken into frames with around 20 ms duration. STFT is applied to the frames to obtain magnitude spectrum. Its spectral envelope reflects the resonance property of the vocal tract, which is closely related to the concept of formants. Usually, the magnitude spectrum is wrapped into perceptually motivated scales such as mel scale, bark scale, and equivalent rectangular bandwidth (ERB) scale. Another Fourier analysis is then taken to derive cepstral features. Among them, MFCC, derived based on mel scale, is the most widely used one. On the other hand, temporal features capture dynamics ignored by short-time features. First-order and second-order delta features fall into this category. Another example is modulation spectral features which represent frequency information of subband signal envelopes (Atlas and Shamma, 2003; Falk and Chan, 2010). They are correlated with speaking rates. Prosodic features capture high level speech information such as rhythm, stress, and intonation (Shriberg et al., 2005; Shriberg, 2007). Although not as discriminative as short-time features, they provide additional information.

Speaker models are built from speaker features. One can simply take an average of all the features and uses it as the model (Campbell, 1997). Obviously such a crude model does not work well. A single Gaussian improves it by incorporating variance of speaker features. However, its capacity is not enough to capture the large variations of speaker features. Vector quantization (VQ) addresses the capacity issue by using a set of feature vectors, known as codewords (Burton, 1987; Soong et al., 1987). During testing, it makes a hard decision for each test feature vector by assigning it to the closest codebook entry. GMM is proposed to combine single Gaussians and VQ (Reynolds, 1995). It replaces
each codeword with a Gaussian function and assigns it with a weight. The mathematical form is written as:

\[
p(X) = \sum_k \pi_k N(X; \mu_k, \sigma_k^2)
\]

where \(k\) is the index of a Gaussian component; \(\mu_k\) and \(\sigma_k\) are the mean and standard deviation of the \(k\)th Gaussian respectively. Typically diagonal covariance matrices are assumed for the Gaussian components. The score of a feature vector \(X\) is the weighted summation of likelihoods from all the Gaussian components. Each Gaussian is considered as modeling a broad phonetic class of the speaker. The GMM framework becomes the predominant approach in speaker recognition due to its simplicity and effectiveness.

For SID, features of a test speech signal are matched with GMMs of all the enrolled speakers. The speaker with the highest score is chosen as the output. For SV, each test utterance has a claimed speaker. There are two hypotheses representing whether the speech is from the claimed speaker or not. A hypothesis test is conducted to derive a likelihood ratio using the GMM of the claimed speaker and the GMM representing everyone else, typically a universal background model (UBM) (Reynolds, 1995; Reynolds et al., 2000). The likelihood ratio is then normalized and compared with a threshold to either accept or reject the original claim. In earlier studies, the GMM of a speaker is trained directly from his/her training data using the expectation-maximization (EM) algorithm. Later this training strategy is replaced by adapting from a pre-trained
UBM. The GMM-UBM framework is proven the better option. Recently, state-of-the-art SV systems extract supervectors using the UBM and further obtain low-dimensional speaker factor or i-vector to deal with channel variation (Kenny, 2006; Dehak et al., 2011).

2.2 Noise-robust Speaker Recognition

Noise robustness of speaker recognition systems has been widely studied over the past decade. Speech enhancement methods such as spectral subtraction (Boll, 1979) have been applied to speaker recognition to alleviate noise corruption. Wang et al. (2011) applied spectral subtraction before extracting MFCC features for SID and SV tasks.

Noise creates a mismatch between clean training data and noisy test data. The idea of introducing noise to training data is pursued. Matsui et al. (1996) modeled speakers and noises separately using hidden Markov models (HMMs) and combined them at different SNR levels. The HMM with the highest likelihood during testing was selected for final decision. They tested it in text-independent SID and SV tasks, and observed considerable performance improvement. On the other hand, Wong and Russell (2001) applied a similar idea to the problem of text-dependent SV. They generated clean HMM-based speaker models and HMM-based noise models, which were combined during recognition using parallel model combination. Ming et al. (2007) proposed to model speakers in multiple noisy training conditions. Models from different training conditions were combined during testing. They adopted missing feature techniques to improve robustness by throwing away heavily corrupted subbands and relying on the rest. Lei et al. (2012)
extracted i-vectors from noisy training data followed by probabilistic linear discriminant analysis (PLDA) to improve robustness in state-of-the-art SV systems.

At the feature level, noise-robust features have been proposed. The European Telecommunication Standards Institute (ETSI) standardized a type of enhanced MFCC features derived from robust speech recognition studies (STQ-AURORA., 2005-11). It employed an advanced front-end (AFE) processing, including voice activity detection and Wiener filtering to improve the robustness of MFCC features. Shao et al. (2007) proposed a noise-robust feature, GFCC, from a gammatone filterbank (see also Shao, 2007). A comparison in noise-robust SID indicates that GFCC is more robust than ETSI-AFE, and both outperform MFCC features (Zhao et al., 2012).

The approaches discussed so far address noise in feature level and model level. Alternative approaches seek to improve noise-robustness with new recognition methods. Missing feature methods, namely marginalization and reconstruction, have produced good results in robust speech recognition (Cooke et al., 2001; Raj et al., 2004). Motivated by these studies, Shao and Wang (2008) applied the missing feature reconstruction method to noise-robust SID coupled with the GFCC features and uncertainty decoding (Srinivasan and Wang, 2007). To identify which T-F units are reliable, they employed a pitch-based monaural speech segregation system to generate CASA masks (Wu et al., 2003). Pullella et al. (2008) exploited the marginalization idea for robust SID. Binary masks are generated using spectral subtraction with further refinement. A mel-filterbank was employed to obtain log-spectral features, which together with the binary masks were fed to a bounded marginalization recognizer. May et al. (2012b) also adopted the bounded marginalization idea but coupled it with the GMM-UBM framework. They
conducted an extensive study on binary mask estimation. Specifically, they estimated noise spectrum and speech spectrum using various state-of-the-art speech enhancement techniques. A comprehensive comparison was made afterwards by supplying the masks to a bounded marginalization recognizer.

### 2.3 Speaker Recognition in Noisy and Reverberant Conditions

Besides noise, reverberation is another common distortion in daily acoustic environments. The combined effects of noise and reverberation have rarely been studied.

Gonzalez-Rodriguez *et al.* (1996) proposed to deal with the combined effects using a low complexity microphone array followed by normalization techniques such as cepstral mean normalization (CMN) and RASTA filtering (Furui, 1981; Hermansky and Morgan, 1994). May *et al.* (2012b) handled the combined effects in a binaural setting. They first performed sound localization using binaural cues and GMM-based classifiers. Then they conducted speech detection to verify whether the detected sound source was speech or noise. The detected speech was supplied to a missing data classifier for SID. Specifically, a binary mask was produced in the speech detection stage to deal with noise. In addition, they refined a channel specific normalization idea developed by Palomaki *et al.* (2004) to handle reverberation.

Both approaches discussed above employed more than one microphone. For monaural speaker recognition, Garcia-Romero *et al.* (2012) dealt with noise and reverberation using multi-conditional training. Specifically, they created noisy training data and reverberant training data. Gaussian PLDA subsystems were trained in each of
the multiple training conditions and combined to produce SV scores. Three training strategies were explored. The first one trained subsystems independently across training conditions. The second one assumed that all subsystems shared the same latent variable. And the last one generated only one set of parameters shared by all subsystems, by pooling all the training data together. They reported results in noisy and reverberant conditions separately. Krishnamoorthy and Prasanna (2009) proposed a system that combined temporal and spectral processing from their earlier study. The temporal processing enhanced time domain signals based on linear predication analysis. On the other hand, the spectral processing estimated a gain function for short-time magnitude spectrum. An enhanced speech spectrum was derived subsequently and combined with noisy phase for a resynthesis of time domain signals, followed by feature extraction and standard speaker recognition. They observed that the combined processing outperformed individual ones in noisy or reverberant conditions. However, they did not report results with the combined effects of noise and reverberation.

Some other methods intend to only address reverberation. De Leon and Trevizo (2007) proposed to train speaker models in multiple reverberant conditions (i.e. rooms) to reduce the mismatch created by reverberation. Multiple models were derived for each speaker and used for SID as if they were from different speakers. Later, they improved the system by incorporating GMM-UBM framework (Akula et al., 2009). Two training strategies were proposed. The first one trained a room-independent (RI) UBM and speaker models were adapted from it. The second one also trained a RI-UBM, and then room-dependent (RD) UBMs were adapted from it using room specific training data. Speaker models were adapted from the RD-UBMs afterwards. During testing, the first
strategy went through all the speaker models and the speaker with the highest score was the output. The second strategy conducted room classification using the RD-UBMs in order to select the closest training room. Then speaker models associated with the selected room were used for SID. A similar reverberation classification idea was adopted by Peer et al. (2008). They trained background models in different reverberant conditions. A reverberation classification was performed to find the best matching background model. The corresponding speaker models were used for SV.

Robust speaker features are studied to combat reverberation. Falk and Chan (2010) proposed modulation spectral features, based on auditory filterbanks. Specifically, input speech was passed through a 23-channel gammatone filterbank. Envelopes of filter outputs were extracted and fed to an 8-channel modulation filterbank to obtain the modulation spectral features, which were shown to be robust to reverberation. Wang and Nakagawa (2009) developed a novel phase extraction method that converted phase into coordinates on a unit circle. They demonstrated that a combination of phase-based features and MFCC was effective in reverberant test conditions.

Borgstrom and McCree (2012) recently proposed a method to enhance reverberant speech. Evaluations on reverberation time estimation and SV produced good performance. Reverberation corresponds to a convolution between an RIR and anechoic speech. With short-time Fourier analysis, the convolution could not be converted to frame level multiplication because typically reverberation time (hundreds of milliseconds) is much longer than the analysis window (20-30 milliseconds). In their study, an assumption was made that the STFTs of the RIR and anechoic speech were convolved with each other. In this case, reverberation was characterized as the channel-wise
convolution of STFTs of the RIR and anechoic speech. The convolution was equivalent to multiplication in the modulation spectral domain after Fourier analysis. The original reverberant speech was enhanced in the modulation spectral domain and resynthesized to time domain for subsequent SV.

An alternative speech enhancement method is blind dereverberation. Sadjadi and Hansen (2014) focused on restoring direct sound and early reflections of reverberant speech as the late reflections were believed to be detrimental. Akin to Wiener filtering, they developed a gain function that was applied to the reverberant spectrum and attenuated the effects of late reflections. Their evaluations indicated that the proposed method worked well in both SID and SV.

### 2.4 Cochannel Speaker Recognition

Cochannel speech is generated when two simultaneous talkers are recorded over a single communication channel. Unlike conversational speech, the two speakers are not aware of each other, resulting in large amounts of overlapping speech. Earlier studies by Yantorno (1998) suggested that speech interference is less detrimental to SID performance than additive noise in a closely controlled experimental setting. He also explored cochannel separation using a proposed harmonic sampling approach. Lovekin et al. (2001) developed the concept of usable speech for SID. Here usable speech refers to the frames minimally corrupted in cochannel speech. It usually happens when one of the two speakers has much higher energy. They extracted usable speech based on frame level target-to-interferer-ratios (TIRs) that were calculated from pre-mixed target and
interference signals. Frames with TIRs higher than a threshold were deemed usable speech. Higher thresholds led to “cleaner” but fewer frames. With a fixed number of usable frames, SID performance improved as the threshold increased. They also explored usable speech selection with spectral autocorrelation ratio.

Shao and Wang (2003) developed a usable speech based cochannel SID system. A multi-pitch tracking system was employed to for usable speech selection (Wu et al., 2003). Single-pitch frames were treated as usable speech and further divided into two groups corresponding to two underlying speakers, based on ground truth pitch information. They evaluated SID performance using two metrics: percentage of one speaker correctly recognized, or percentage of the target speaker correctly recognized. Later they extended the system to jointly perform cochannel SID and separation (Shao and Wang, 2006a). The multi-pitch tracking system was again used as a front-end to obtain usable frames, which were grouped into segments based on pitch continuity. A speaker assignment variable was introduced to distribute usable segments to the two speakers. They searched all the speaker pairs and assignments to find the optimal speaker pair and assignment. They proposed a hypothesis pruning algorithm to reduce the time complexity of the search. Shao et al. (2010) further improved the system by incorporating CASA-based speech separation. Voiced speech was first segregated using cross channel correlation and temporal continuity, and then grouped into simultaneous streams (Hu and Wang, 2006). Subsequently the simultaneous streams were supplied to the joint SID and separation system for sequential grouping, resulting in two streams corresponding to two speakers. Missing feature reconstruction with uncertainty decoding was employed for
SID scoring during the search. Unvoiced speech segregated using an onset/offset based system (Hu and Wang, 2007) was later merged into the two streams.

Hershey et al. (2010) developed a multi-talker speech recognition system. It adopted a cochannel SID system that produced the state-of-the-art performance on the SSC corpus. The SID system has two stages. The first stage creates a short list of most probable speaker candidates. Specifically, it calculates posterior probabilities of speakers given each frame, followed by an aggregation across frames. Top 6 speakers are selected for the second stage, where the top speaker is paired with each of the rest. An EM algorithm is applied to each speaker pair to estimate the gains. The speaker pair whose gain-adapted models generated the highest utterance level likelihood is regarded as the output. Guan and Liu (2008) proposed a similar system that was incorporated by Li et al. (2010) in their participating system in an SSC contest. The first stage is almost the same with Hershey et al.’s system except for some differences in score calculation and aggregation of posterior probabilities. Besides, top 10 speakers are kept instead of 6. Differences lie in the second stage, where the top speaker model is combined with each of the other models for SID. The pair whose combined model generates the highest likelihood is the output. This system achieved the state-of-the-art performance with an average performance of around 99%, 1 percent higher than that of Hershey et al.’s system.
2.5 Speaker Recognition with Channel Distortions

Speech is often recorded over different microphones and telephone channels. Different channels distort speech spectrum differently. Training and testing in different channels face a mismatch problem.

Channel distortions are modeled as a convolution between a speech signal and a channel filter. Unlike reverberation, the length of the filter is usually smaller than the short-time Fourier analysis window. The distortion becomes an additive term in log-spectral domain or cepstral domain, and thus can be removed by spectral mean subtraction or CMN. An underlying assumption made here is that channel characteristics are constant over time. To relax this assumption, one can normalize features within each segment (Viiikki and Laurila, 1998). Hermansky and Morgan (1994) proposed RASTA (relative spectra) to remove slow varying convolutive channel noise. The RASTA filter is applied to features vectors along the time dimension.

Teunen et al. (2000) developed speaker model synthesis to deal with channel mismatch. They first created a number of channel dependent background models adapted from a channel independent root model. Then they learned transformations among the channel dependent background models. For a specific speaker, the best matching channel condition is identified and the corresponding background model is adapted to get the speaker model. Speaker models of other channel conditions are synthesized based on the learned transformations. During testing, the best matching channel is identified using channel dependent background models and the corresponding speaker model is used for SV.
Reynolds (2003) proposed a similar approach called feature mapping. He also adapted channel dependent background models from a channel independent root model. The relationships between the adapted models and the root model are recorded. Given a speech signal, the best matching channel dependent model is detected and the corresponding relationship with the root model is used to map features to the channel independent space.

Supervector is widely used in recent speaker recognition studies. It is a high dimensional feature vector that can be derived in different ways. A commonly used method simply stacks all the means of a UBM to get the supervector. An advantage of the supervector is that it maps utterances with varying lengths to features of the same dimension. Channel distortions in the supervector can be compensated for via nuisance attribute projection (Solomonoff et al., 2004), linear discriminant analysis, or within-class covariance normalization (Hatch and Stolcke, 2006). The processed supervector can subsequently be fed to SVMs for speaker verification.

Kenny (2006) explicitly models channel distortions using JFA. A speaker’s supervector $M$ is decomposed as follows:

$$M = m + V y + U x + D z$$  \hspace{1cm} (2.2)

where $m$ denotes a speaker and channel independent component from the UBM, $V$ a speaker dependent eigenvoice matrix defining a speaker space, and $y$ a speaker factor. $U$ is an eigenchannel matrix corresponding to a channel space, and $x$ is a channel factor. $D$ and $z$ together define a speaker dependent residual space. JFA first estimates all the
matrices from labeled training data, and then factors. Channel compensation is achieved by discarding the channel dependent part of $Ux$.

Dehak et al. (2011) proposed to model only one total variability space $T$, instead of multiple spaces in (2.2):

$$M = m + Tw$$

(2.3)

where $m$ is the same as in (2.2) and $T$ the total variability space. Here $w$ is the identity factor, also known as $i$-vector. This formulation is motivated by Dehak’s findings that the channel factor $x$ in JFA also contains speaker specific information (Dehak, 2009). SVM performance using the i-vector is shown to outperform SVM-based and JFA-based approaches, and regarded as the state-of-the-art.

### 2.6 Summary

Previous sections reviewed how existing approaches deal with different kinds of distortion. In the next three chapters, we will present our proposed methods for noise-robust SID, dealing with the combined effects of noise and reverberation, and addressing cochannel SID in both anechoic and reverberant conditions. Note that this dissertation does not address channel distortion.
CHAPTER 3

NOISE-ROBUST SPEAKER IDENTIFICATION

3.1 Introduction

In this chapter, we propose a robust SID system by using CASA as a front-end to perform speech segregation. The output of CASA segregation is in the form of a binary T-F mask. We first introduce speaker features, GF and GFCC, based on an auditory periphery model (Shao and Wang, 2007, 2008). Specifically, a GF is first obtained from a bank of gammatone filters. Then, GFCC is derived from GF by a cepstral analysis. We show that GFCC achieves an SID level of performance in noisy environments that is significantly better than MFCC. The proposed system has two modules. To account for the deviations of noisy features from clean ones, the first module enhances the GF by reconstructing corrupted components indicated by a CASA-generated binary T-F mask. The second module performs bounded marginalization on the noisy GF. Each module yields substantial improvement over baseline SID systems. As the two modules perform well in different conditions, we propose a combined system integrating these two modules.
The rest of the chapter is organized as follows. Section 3.2 describes the overall system architecture. Auditory feature extraction and binary mask estimation are discussed in Section 3.3. Section 3.4 and 3.5 introduce the reconstruction module and the marginalization module, respectively. The two modules are combined in Section 3.6. SID evaluations and comparisons are presented in Section 3.7. Further discussions are given in Section 3.8. The results in this chapter have been published in Zhao et al. (2011, 2012) and Zhao and Wang (2013).

### 3.2 System Overview

The proposed system uses CASA as a front-end processor for robust SID. Fig. 3.1 presents the diagram of the overall system. Input speech is decomposed using a gammatone filterbank and subsequent time windowing to generate a time sequence of GFs. This T-F analysis results in a cochleagram (Wang and Brown, 2006), which is a two-dimensional representation of the input signal. Simultaneously, we feed the input signal to a CASA system that computes a binary mask corresponding to the target speech (Jin and Wang, 2009). Elements of this mask correspond to T-F units in the cochleagram, with 1 indicating that the corresponding T-F unit is dominated by target and 0 by noise. The binary mask and GFs are fed to both the reconstruction module and the marginalization module.
In the reconstruction module, the noise-corrupted components indicated by the CASA mask are reconstructed using a speech prior (Raj et al., 2004) and the enhanced GF is converted to the cepstral domain by discrete cosine transform (DCT). Subsequently, the obtained cepstral feature, GFCC, is used in conjunction with trained speaker models to derive the underlying speaker identity. In the marginalization module, there is no need for missing feature reconstruction. Bounded marginalization is performed on the noisy GF directly with the CASA mask providing the information of which T-F units are corrupted and hence marginalized.

Each module provides an SID system by itself. Our experiments suggest that the reconstruction module and the marginalization module work well in different conditions. To leverage their respective advantages, our combined system assigns the input signal to both modules and integrates the individual outputs to make the final decision. Note that the two modules as well as the combined system operate on a per utterance basis.

Figure 3.1: Schematic diagram of a CASA-based robust speaker identification system.
3.3 Feature Extraction and Mask Estimation

In this section, we describe how to extract GF and GFCC features from the cochleagram, and compute a CASA mask.

3.3.1 Auditory Features

Our system first performs auditory filtering by decomposing an input signal into the T-F domain using a bank of gammatone filters. Gammatone filters are derived from psychophysical and physiological observations of the auditory periphery and this filterbank is a standard model of cochlear filtering (Patterson et al., 1992). We use a bank of 64 filters whose center frequencies range from 50 Hz to 4000 Hz or 8000 Hz depending on the sampling frequency of speech data. Since the filter output retains the original sampling frequency, we decimate fully rectified 64-channel filter responses to 100 Hz along the time dimension. This yields a corresponding frame rate of 10 ms, which is used in many short-time speech feature extraction methods. The magnitudes of the decimated outputs are then loudness-compressed by a cubic root operation:

\[ G_m[i] = \left\| g_{\text{decimate}}[i, m] \right\|^{1/3}, \quad i = 0 \ldots N - 1, \quad m = 0 \ldots M - 1 \]  

(3.1)

Here, \(N=64\) refers to the number of frequency (filter) channels. \(M\) is the number of time frames obtained after decimation. The resulting responses \(G_m[i]\) form a matrix, representing the T-F decomposition of the input. This T-F representation is a variant of cochleagram. Note that, unlike the linear frequency resolution of a spectrogram, a
cochleagram provides a finer frequency resolution at low frequencies than at high frequencies. Fig. 3.2 shows a cochleagram and a spectrogram of an utterance. Darker regions represent stronger energy. Note the difference in energy-concentrated regions below 1000 Hz between these two T-F representations. We base our subsequent processing on the cochleagram representation.

![Cochleagram and Spectrogram](image)

Figure 3.2: Illustrations of a cochleagram (top) and a spectrogram (bottom) of a clean speech utterance. Note the asymmetric frequency resolution at low and high frequencies in the cochleagram.

Each time slice of the above matrix is GF, and $G[i]$ denotes its $i$th channel. Time index $m$ is dropped for simplicity. Here, a GF vector comprises 64 frequency components. Note that the dimension of a GF vector is larger than that of MFCC vectors used in a typical speaker recognition system. Additionally, because of the frequency overlap among neighboring filter channels, GF components are correlated with each other. In order to reduce dimensionality and de-correlate the components, we apply a
DCT to a GF. The resulting coefficients are GFCC (Shao et al., 2007; Shao and Wang, 2008). Specifically, cepstral coefficients, $C[j]$, $j=0...N-1$, are obtained from a GF as follows,

$$C[j] = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} G[i] \cos \left( \frac{j \pi}{2N} (2i + 1) \right), \quad j = 0...N - 1$$ (3.2)

Note that the 0th order coefficient summates all the GF components. Thus, it relates to the energy of a GF vector.

Rigorously speaking, the newly derived coefficients are not cepstral coefficients because a cepstral analysis requires a log operation between the first and the second frequency analysis for the deconvolution purpose (Oppenheim et al., 1999). Here we call them cepstral coefficients because of the functional similarities between the above transformation and that of a typical cepstral analysis in the derivation of MFCC.

### 3.3.2 CASA-based Mask Estimation

As described earlier, a cochleagram is a T-F representation of a signal. With such a representation, a binary T-F mask furnishes the crucial information about whether a T-F unit is dominated by target speech or background noise.

As a main computational goal of CASA, an ideal binary mask (IBM) is a binary matrix defined as follows (Wang, 2005):
\[ IBM(t, f) = \begin{cases} 
1, & \text{if } SNR(t, f) > LC \\
0, & \text{otherwise} 
\end{cases} \]  \hspace{1cm} (3.3)

\( SNR(t, f) \) refers to the local SNR of the T-F unit in time frame \( t \) and frequency channel \( f \). \( LC \) denotes an SNR threshold called \emph{local criterion}. In this chapter, \( LC \) is set to 0 dB. Given premixed target and interference signals, the IBM can be readily constructed. The IBM concept is motivated by the auditory masking phenomenon (Moore, 2003) and is the optimal binary mask in terms of SNR gain (Li and Wang, 2009).

To estimate the IBM from an input mixture, we employ a recent CASA system that performs feature-based classification (Jin and Wang, 2009). First, we estimate the pitch of the speech signal at each frame using a multi-pitch tracking algorithm (Jin and Wang, 2011a). This algorithmformulates multi-pitch tracking as a HMM, which can produce up to 2 pitch points at each frame. As we deal with noises that are mostly aperiodic, the multi-pitch tracker tends to output at most one pitch per frame. Given an estimated pitch, a 6-dimensional pitch-based feature vector is extracted for each T-F unit. These features are fed to an MLP classifier, whose output can be interpreted as the posterior probability of a T-F unit being target dominant. The desired output during MLP training is the IBM. Note that we take the binarized MLP output as the resulting CASA mask without using a subsequent segmentation and grouping stage in the original system of Jin and Wang (2009).
Fig. 3.3 shows an estimated IBM for a noisy speech utterance. If input SNR is given, an SNR-dependent MLP can be trained to estimate the IBM. Otherwise one can train multiple MLPs at different SNRs, and select the MLP whose corresponding SNR is closest to the estimated input SNR. In this chapter, the latter is adopted as we assume no prior knowledge of input SNR. More details about MLP training will be provided in Section 3.7.1.
3.4 Reconstruction Module

In speaker recognition, the probability distribution of an extracted feature vector $X$ produced by a speaker $\lambda$ is typically modeled as a GMM (Reynolds, 1995), parameterized by diagonal covariance matrices. Under noisy conditions, the aforementioned speech segregation system produces a binary T-F mask that indicates whether a GF component is speech dominant or noise dominant. The former one is regarded as reliable since the system has more speech information in the speaker models while the latter one is deemed missing. Thus, the feature vector is partitioned into reliable components $X_r$, and unreliable ones $X_u$,

$$X = \begin{bmatrix} X_r \\ X_u \end{bmatrix}$$

In order to enhance a noise corrupted GF, we first reconstruct its missing components from a speech prior model, which is similar to a UBM in speaker verification (Reynolds, 1997). Specifically, the speech prior $p(X)$ is modeled as a large GMM (Raj et al., 2004), and obtained from pooled training data:

$$p(X) = \sum_{k=1}^{K} p(k) p(X|k)$$  \hspace{1cm} (3.5)

where $K$ is the number of mixture components, and $k$ denotes the index and $p(k)$ the prior probability of a mixture component. $p(X|k)$ is the $k$th Gaussian distribution with a mean
vector $\mu_k$ and a diagonal covariance $\Sigma_k^2$. Given a binary mask, the components of the mean and variance of each Gaussian are also split into reliable and unreliable ones. We then calculate the \textit{a posteriori} probability of the $k$th component given reliable GF components as

$$p(k \mid X_r) = \frac{p(k)p(X_r \mid k)}{\sum_{k=1}^{K} p(k)p(X_r \mid k)}$$

(3.6)

As shown in Cooke \textit{et al.} (2001) and Srinivasan and Wang (2007), the unreliable components are estimated as the expected value or the mean conditioned on $X_r$.

$$\hat{X}_u = \sum_{k=1}^{K} p(k \mid X_r) \mu_{u,k}$$

(3.7)

where $\mu_{u,k}$ refers to the mean vector of the unreliable components of the $k$th Gaussian in the speech prior. The reliable components are retained in the reconstruction. Since GF is an energy-based feature, the underlying target signal is expected to be smaller than the mixture value. Therefore, we replace a reconstructed value with the observed value if the former is larger.

As shown in the above equations, the quality of reconstruction is largely determined by the amount of reliable speech information. With little reliable information, the quality of recognition is expected to be very poor. Therefore, we introduce a frame selection step
in the reconstruction module to choose relatively clean frames, when there are plenty frames available for recognition. Some criterion such as frame level SNR or the number of reliable units is needed for selection, and details will be provided in Section 3.7.4.

With the reconstructed GF, we convert it into GFCC by applying DCT. GFCC is a speaker feature that can be directly used for recognition in conjunction of trained speaker models as described in Section 3.3.1.

### 3.5 Marginalization Module

An alternative approach to reconstruction is marginalization, which has shown good performance in robust speech recognition (Cooke et al., 2001) and has been applied to robust speaker recognition (El-Maliki and Drygajlo, 1999; Shao and Wang, 2006b). The main idea behind marginalization is to base recognition decisions on reliable components; in other words, we want to marginalize unreliable components. With GMM speaker models and diagonal covariance matrices, we have

\[
p(X_r|\lambda) = \int_{-\infty}^{\infty} p(X_r, X_u|\lambda) dX_u \\
= \int_{-\infty}^{\infty} \sum_{k=1}^{K} p(k)p(X_r, X_u|k) dX_u \\
= \sum_{k=1}^{K} p(k)p(X_r|k) \int_{-\infty}^{\infty} p(X_u|k) dX_u \\
= \sum_{k=1}^{K} p(k)p(X_r|k) \\
\]

(3.8)
In the above equation, an unreliable feature dimension integrates to 1 and the likelihood calculation reduces to a simple case where the feature dimensions of reliable T-F units are inserted into each speaker model to get the likelihood of a frame.

Although from the unreliable T-F units we cannot precisely predict the underlying target feature value, the feature value should be within the range from 0 to the observed value as a GF feature is derived from the cubic root operation (see (3.1)). This analysis provides a more accurate range of integration than that from minus infinity to positive infinity in (3.8). Utilizing the tighter range leads to bounded marginalization (Cooke et al., 2001), described below where “low” and “high” define the range:

\[
p(X_r | \hat{\lambda}) = \int_{low}^{high} p(X_r, X_u | \hat{\lambda}) dX_u \\
= \sum_{k=1}^{K} p(k) \int_{low}^{high} p(X_r, X_u | k) dX_u \\
= \sum_{k=1}^{K} p(k) \int_{low}^{high} p(X_r | k) p(X_u | k) dX_u \tag{3.9}
\]

Consistent with earlier studies (El-Maliki and Drygajlo, 1999; Cooke et al., 2001), we have found that bounded marginalization produces substantially better recognition performance than full marginalization. Therefore, we employ bounded marginalization on GF features. It is worth emphasizing that this marginalization method operates in the spectral domain, whereas the reconstruction method described in Section 3.4 performs recognition in the cepstral domain.
3.6 Combined System

Between the reconstruction module and the marginalization module, we expect the former to perform better at high SNRs as it is well known that cepstral features outperform spectral features in recognition (Davis and Mermelstein, 1980; Srinivasan et al., 2006). On the other hand, marginalization is expected to perform better in low SNR conditions, as reconstruction based on few reliable T-F units likely has poor quality. Also, bounded marginalization makes use of some information from unreliable T-F units. These differing performance trends are indeed confirmed by the evaluation results presented in the next section. To utilize the relative advantages, we combine them into one system.

In our study, we have noticed that when a module makes a recognition mistake, the underlying target speaker tends to have a top ranked score although it is not the highest. Meanwhile, wrong identities from these two modules tend not to agree. Motivated by this observation, we simply use a linear combination.

We derive an SID score vector for each frame by feeding the frame signal to each speaker model. Note that each element of this vector is a log-likelihood corresponding to a particular speaker model. An utterance level score vector is derived by adding frame level log-likelihood score vectors. After integrating SID scores from all the available frames, each module outputs a score vector with the number of elements equal to the total number of the speaker models. As the scores from the two modules may not be on the same scale, normalization should be applied before adding them together. We perform the following simple normalization:
\[
\hat{s}_{\text{Module}}(\lambda) = \frac{s_{\text{Module}}(\lambda) - \min_\lambda (s_{\text{Module}}(\lambda))}{\max_\lambda (s_{\text{Module}}(\lambda)) - \min_\lambda (s_{\text{Module}}(\lambda))}
\]  

(3.10)

where \(s_{\text{Module}}(\lambda)\) and \(\hat{s}_{\text{Module}}(\lambda)\) denote the original and normalized score vectors of an individual module respectively. The SID score of the combined system is given below:

\[
s(\lambda) = \hat{s}_{\text{REC}}(\lambda) + \hat{s}_{\text{MAR}}(\lambda)
\]  

(3.11)

We refer to the frames containing at least one reliable T-F unit as “active frames”. The frames containing no reliable unit are either unvoiced speech mixed with noise or voiced speech completely masked by noise. Our study shows that unvoiced speech plays a relatively minor role in speaker recognition and our CASA-mask estimation algorithm cannot separate unvoiced speech. Completely masked voiced speech provides little information for SID and it seems reasonable to ignore these frames. Therefore, we only feed active frames to the two modules.

### 3.7 Evaluation and Comparison

In this section, we systematically evaluate the noise robustness of the proposed SID methods. We also compare the performance of our system with baseline systems using the conventional MFCC feature and the ETSI-AFE feature. In addition, we compare with a related robust SID system by Pullella et al. (2008).
3.7.1 Experiment Setup

We employ speech material (one-speaker detection, cellular data) from the 2002 National Institute of Standards and Technology (NIST) Speaker Recognition Evaluation (SRE) corpus (Przybocki and Martin, 2002), which is a standard dataset for automatic speaker recognition (particularly verification). The speaker dataset contains 330 speakers. Each speaker has a roughly 2-minute long telephone recording sampled at 8 kHz for training. It is divided into 5s long pieces, and 2 of them are included in the test set, 2 in the development set and the remaining ones in the training set. To study how the proposed system performs under different types of noisy conditions, the test utterances are mixed with multitalker babble noise which is nonstationary, speech shape noise (SSN, stationary), and factory noise (nonstationary). Each noise is mixed with telephone speech at various SNR levels from −6 dB to 18 dB at 6 dB intervals. Note that the test utterances are different from the training ones.

Sixty-four dimensional GF is extracted to model speaker dependent characteristics. To reconstruct the noisy GF, a speech prior with 2048 Gaussian components is trained using all the pooled training data. The reconstructed GF is converted to GFCC using DCT. Each speaker model is adapted from a 1024-component UBM trained using all the training data (Reynolds et al., 2000). Compared with individually trained GMMs, this GMM-UBM approach scores much faster and is more discriminative.

As the NIST dataset contains telephone speech, little speech information exists below 200 Hz. Therefore we only use features above 200 Hz. In the gammatone filterbank, the 10 lowest channels correspond to frequencies below 200 Hz and thus GF consists of
channels 11-64 (i.e. 54 channels). As confirmed using the development set, excluding the low-frequency channels increases SID performance.

For MLP training, we randomly select 50 utterances from the training set and mix them with speech shape, factory, babble and white noises at SNR levels from −12 dB to 18 dB with 6 dB increment. At each SNR, an SNR-specific MLP is trained. In addition, a generic MLP is trained by pooling mixtures from all SNR levels. Given a test speech signal, the generic MLP is used to generate a binary mask, from which we estimate the input SNR (during voiced intervals). For separation, we choose the MLP whose training SNR is closest to the estimated SNR.

3.7.2 GFCC Dimensions and Dynamic Features

In the reconstruction module, when converting 64-dimensional GF to GFCC, keeping all the 64 dimensions of GFCC may not be necessary. After inverting DCT of GFCC, we find that the lower 23-order coefficients capture almost all the GF information and the coefficients above the 23th have values close to 0, which means that they provide negligible information (see also Shao and Wang, 2008). As an illustration, Fig. 3.4(a) shows the cochleagram of an utterance, Fig. 3.4(b) shows a comparison of a GF frame at 1 sec of Fig. 3.4(a) and the resynthesized GF from the first 23 GFCC coefficients, and Fig 3.4(c) presents the resynthesized cochleagram from the top plot using only the 23 coefficients. As can be seen from the figure, the lower 23-order GFCCs largely retain the information in 64-dimensional GFs. This is due to the “energy compaction” property of DCT (Oppenheim et al., 1999). Additionally, the 0th cepstral coefficient corresponds to
the energy of the whole frame, which is susceptible to noise corruption. Our experiments using the IBM for separation show that removing the 0\textsuperscript{th} coefficient improves the SID performance significantly. Hence, in the later experiments we will use 22-dimensional GFCCs.

![Figure 3.4: Illustrations of energy compaction by GFCCs. Plot (a) shows a cochleagram of an utterance. Plot (b) shows a GF frame at 1 sec of (a). The original GF is plotted as the solid line and the resynthesized GF by 23 GFCCs is plotted as the dashed line. Plot (c) presents the resynthesized cochleagram from (a) using 23 GFCCs.](image-url)
Since a typical speaker recognition system uses MFCCs and their first-order (delta) dynamic coefficients, it is reasonable to study how GFCC dynamic features fare for recognition. GFCCs with 22 dimensions have shown good SID performance in our experiments. After appending 22-dimensional dynamic features, we find that the performance improvement is not significant. Therefore, we use 22-dimensional static GFCCs as speaker features in the reconstruction module.

### 3.7.3 Baseline Comparisons

To show the utility of GFCC as speaker features, we choose MFCC and ETSI-AFE as baseline features. ETSI-AFE is essentially enhanced MFCC features. Our experiments suggest that MFCC without delta or acceleration features performs better. This is probably because without noise reduction, the delta and acceleration features are very noisy and cannot encode the underlying dynamic speaker information. However, ETSI-AFE with delta features works better than static features. Therefore, we choose static MFCC features and ETSI-AFE with delta features as two baselines. For the GFCC baseline, we directly derive noisy GFCC features out of a mixture without separation or reconstruction. In this way, we could directly evaluate the effectiveness of GFCC as a new speaker feature. To make a fair comparison, since the GFCC feature has 22 dimensions, we also derive 22-dimensional MFCCs in addition to the commonly used 12-dimensional version (after removing the 0\textsuperscript{th} coefficient). As mentioned in Section 3.7.1, we only use GF features above 200 Hz. This is also the case for MFCC features. As for
ETSI-AFE features, we use the default frequency range as it is unclear how to adjust the frequency range.

Figure 3.5: SID performance of different baseline systems for 3 noises.

Fig. 3.5 gives the SID accuracies of different baseline systems with respect to SNR. When performing SID, we only consider active frames. The results in the figure show that the GFCC baseline on average gives significantly better performance than the other baselines for all three noises. This indicates that the GFCC feature has more robustness in
noisy conditions. ETSI-AFE_D (_D indicates delta features) works better than 12-dimensional MFCC features. After we increase MFCC dimensions to 22, the same dimensionality as GFCC, MFCC features yield closer performance to ETSI-AFE_D but still underperform GFCC features. We will analyze the robustness differences of GFCC and MFCC features in Section 3.8.

3.7.4 Evaluation Results

Now we present SID results of the proposed methods using estimated IBM. We also compare the performance of the individual modules and the combined system. In the frame selection step of the reconstruction module, we use as the selection criterion the smaller of half of the frequency channels (i.e. 27 for the NIST dataset – see Section 3.7.1) and the median number of reliable T-F units of all active frames for a noisy speech utterance. Given an active frame, it will be selected if its number of reliable units is greater than the criterion.

Table 3.1 shows the SID performances of the three methods: the reconstruction module, the marginalization module, and the combined system. As shown in the table, at high SNR conditions, particularly at 18 dB, the reconstruction module with GFCC performs well, better than GF plus bounded marginalization that operates in the spectral domain. On the other hand, the marginalization module performs consistently better under low SNR conditions. This suggests that, when there are relatively many reliable T-F units, reconstructing unreliable ones and using GFCC features yield performance advantages. On the contrary, if there are few reliable T-F units, bounded marginalization
in the spectral domain is a more effective strategy. We should point out that, in terms of computational complexity, the reconstruction module is faster as it uses 22-dimensional GFCC features, as opposed to 54-dimensional GF features used in the marginalization module. Also, the integration operation in bounded marginalization (see (3.9)) takes time. These factors lead to the reconstruction module taking about 1/3 of the computing time of the marginalization module.

<table>
<thead>
<tr>
<th>Babble</th>
<th>−6dB</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFCC_REC</td>
<td>13.94</td>
<td>47.58</td>
<td>70.45</td>
<td>79.85</td>
<td>86.21</td>
<td>59.61</td>
</tr>
<tr>
<td>GF_MAR</td>
<td>21.21</td>
<td>54.85</td>
<td>72.12</td>
<td>80.15</td>
<td>83.79</td>
<td>62.42</td>
</tr>
<tr>
<td>Combined System</td>
<td>31.97</td>
<td>70.30</td>
<td>82.12</td>
<td>87.88</td>
<td>90.61</td>
<td>72.58</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factory</th>
<th>−6dB</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFCC_REC</td>
<td>19.55</td>
<td>46.21</td>
<td>68.94</td>
<td>78.48</td>
<td>86.52</td>
<td>59.94</td>
</tr>
<tr>
<td>GF_MAR</td>
<td>28.64</td>
<td>53.18</td>
<td>65</td>
<td>69.24</td>
<td>81.21</td>
<td>59.45</td>
</tr>
<tr>
<td>Combined System</td>
<td>38.33</td>
<td>67.73</td>
<td>77.88</td>
<td>83.64</td>
<td>89.09</td>
<td>71.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSN</th>
<th>−6dB</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFCC_REC</td>
<td>15.76</td>
<td>40.76</td>
<td>66.21</td>
<td>78.64</td>
<td>85.76</td>
<td>57.43</td>
</tr>
<tr>
<td>GF_MAR</td>
<td>25.91</td>
<td>55</td>
<td>72.73</td>
<td>77.73</td>
<td>82.12</td>
<td>62.7</td>
</tr>
<tr>
<td>Combined System</td>
<td>29.85</td>
<td>67.42</td>
<td>82.58</td>
<td>86.97</td>
<td>89.09</td>
<td>71.18</td>
</tr>
</tbody>
</table>

Table 3.1: SID accuracy (%) of the proposed methods. GFCC_REC denotes the reconstruction module. And GF_MAR the marginalization module.

The combined system attempts to take advantage of the two methods. By looking at the SID results in Table 3.1, it is clear that on average the marginalization module works
better than reconstruction module for babble and SSN. The combined system significantly outperforms the individual modules.

<table>
<thead>
<tr>
<th>Babble</th>
<th>−6dB</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFCC_REC</td>
<td>27.58</td>
<td>50</td>
<td>71.06</td>
<td>81.21</td>
<td>88.94</td>
<td>63.76</td>
</tr>
<tr>
<td>GF_MAR</td>
<td>69.85</td>
<td>77.73</td>
<td>85</td>
<td>88.48</td>
<td>90.30</td>
<td>82.27</td>
</tr>
<tr>
<td>Combined System</td>
<td>60.91</td>
<td>79.70</td>
<td>87.27</td>
<td>91.97</td>
<td>93.33</td>
<td>82.64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factory</th>
<th>−6dB</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFCC_REC</td>
<td>21.97</td>
<td>47.42</td>
<td>68.94</td>
<td>81.06</td>
<td>87.27</td>
<td>61.33</td>
</tr>
<tr>
<td>GF_MAR</td>
<td>51.97</td>
<td>70.30</td>
<td>80.45</td>
<td>86.06</td>
<td>87.58</td>
<td>75.27</td>
</tr>
<tr>
<td>Combined System</td>
<td>46.36</td>
<td>73.03</td>
<td>85</td>
<td>91.52</td>
<td>92.73</td>
<td>77.73</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSN</th>
<th>−6dB</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFCC_REC</td>
<td>19.70</td>
<td>43.94</td>
<td>66.21</td>
<td>79.39</td>
<td>86.36</td>
<td>59.12</td>
</tr>
<tr>
<td>GF_MAR</td>
<td>55.15</td>
<td>74.70</td>
<td>82.27</td>
<td>87.73</td>
<td>89.24</td>
<td>77.82</td>
</tr>
<tr>
<td>Combined System</td>
<td>52.27</td>
<td>75.30</td>
<td>85.61</td>
<td>91.36</td>
<td>93.18</td>
<td>79.54</td>
</tr>
</tbody>
</table>

Table 3.2: SID accuracy (%) of the proposed methods with the IBM.

To evaluate the quality of IBM estimation, we present the SID performance using the IBM in Table 3.2. The table shows that both modules work very well using the IBM, especially the marginalization module. Compared with Table 3.1, the reconstruction module has less significant improvement than the marginalization module. This may reflect the robustness of the reconstruction module to mask estimation errors. The dramatic gap between the two modules at −6 dB leads to a little performance degradation.
in the combined system. However, at 0 dB, although the gap is still large, the combined system is able to further improve the individual results.

Equation (3.11) weights the two modules equally. This combination is very simple, and it is possible that using unequal weights, e.g. assigning a higher weight to the more accurate module, produces better identification results. In the above IBM evaluation, we have found that, when we weight the two modules proportional to the numbers of selected frames, the performance of the combined system is improved a little compared to (3.11) as marginalization uses more active frames and therefore contributes more to the combination.

Table 3.3 lists the average SID results of the combined system along with those of the baseline systems given in Section 3.7.3. Clearly the combined system outperforms all three baselines. The combined system’s SID results are more than 28 percentage points higher than those of MFCC and ETSI-AFE_D baselines. The gain over the GFCC baseline is smaller, reflecting the robustness of GFCC features themselves.

<table>
<thead>
<tr>
<th>Method</th>
<th>Babble</th>
<th>Factory</th>
<th>SSN</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined System</td>
<td>72.58</td>
<td>71.33</td>
<td>71.18</td>
<td>71.7</td>
</tr>
<tr>
<td>GFCC_22</td>
<td>47.64</td>
<td>51.46</td>
<td>49.61</td>
<td>49.57</td>
</tr>
<tr>
<td>MFCC_22</td>
<td>39.42</td>
<td>35.95</td>
<td>31.58</td>
<td>35.65</td>
</tr>
<tr>
<td>MFCC_12</td>
<td>35.27</td>
<td>29.7</td>
<td>26.55</td>
<td>30.51</td>
</tr>
<tr>
<td>ETSI-AFE_D</td>
<td>40.55</td>
<td>45.33</td>
<td>43.27</td>
<td>43.05</td>
</tr>
</tbody>
</table>

Table 3.3: SID accuracy (%) of the combined system and baselines. Performance is averaged across different SNR conditions.
Under clean conditions, MFCC_22 yields the SID accuracy of 96.67% (94.39% for MFCC_12), whereas the accuracy is 97.12% for GFCC_22. GF as a spectral feature gives the accuracy of 95.76%, which is slightly worse than the 22-dimensional cepstral features.

In a similar task on the 2002 NIST dataset, the accuracy of 89.39% was reported on the clean test set using MFCC features (Apsingekar and De Leon, 2009).

### 3.7.5 Comparison with a Related System

Pullella et al. (2008) recently proposed a system for robust speaker recognition, which also utilizes bounded marginalization to achieve noise robustness. The difference from our marginalization module lies in two aspects. First, we use the gammatone filterbank as the front-end followed by decimation to derive GF features. They use a mel-scale filterbank as the front-end. The second difference is in mask estimation. They compute a binary mask using spectral subtraction, and then feature selection to refine the initial mask. It is questionable whether spectral subtraction can effectively deal with nonstationary noises. As described earlier, our system uses CASA-based speech segregation to directly estimate the IBM.

Our comparison uses the same experimental setup as in Pullella et al.’s system. The speech signals are from the TIDigits corpus (Leonard, 1984), from which 31 speakers (21 males and 10 females) are randomly chosen. Each speaker has speech utterances corresponding to 77 connected digits. Out of them, 50 are randomly chosen for training and 27 for testing. Test utterances are corrupted by white noise and factory noise at −5, 0, 5, 10, 15, and 20 dB. In the following figures, the performance of their system and
MFCC baseline is directly taken from Pullella et al. It is worth mentioning that in this simulation we use individually trained GMMs instead of the GMM-UBM scheme to be consistent with their system, and the frame selection step is not employed due to relative short test utterances. The mask estimation process is the same as described in Section 3.7.1 except that babble, white, factory and destroyer (operation room) noises are employed for MLP training.

Figure 3.6: SID accuracy (%) comparisons of the proposed marginalization module and Pullella et al.’s system. Both systems utilize the ideal binary mask for separation.

Fig. 3.6 shows the SID performance with the IBM. To sharpen the comparison, we give the performance of the marginalization module of our system. The figure shows that our marginalization module yields SID accuracies that are about 10 percentage points higher than those in Pullella et al. for both white noise and factory noise conditions. In our system, only active frames are used for recognition, while their system appears to use all the frames. In this case, our system using all the frames
achieves almost the same performance as using active frames. Therefore this improvement should reflect the relative advantage of GF features over their mel-scale features.

Fig. 3.7 shows the SID performances of the proposed methods and Pullella et al.’s system with their respective methods of mask estimation. The comparison shows that all of our proposed methods perform much better than their system in both noise conditions, particularly at lower SNR levels. While our methods’ performance does not vary a lot for the two noises, their system performs considerably worse in the factory noise, presumably because of the ineffectiveness of spectral subtraction for attenuating this nonstationary noise.

Figure 3.7: SID accuracy (%) comparisons of the proposed combined system and Pullella et al.’s system using estimated binary masks.
Comparing Fig. 3.6 and Fig. 3.7, our system with estimated binary masks does not degrade the performance by much compared to the use of the IBM, unlike the performance gaps in the NIST corpus shown earlier (cf. Table 3.1 and Table 3.2). We believe that this can be attributed to the large lexicon overlap between training and testing in the TIDigits corpus, which has a very small vocabulary. In the NIST corpus, there is no overlap between training and test utterances. We will come back to this point in Section 3.9.

3.8 Analyzing Noise Robustness of MFCC and GFCC Features

Why does GFCC appear to be more robust than MFCC (see Section 3.7.3), at least for speaker identification? We believe this is an important question for the study of noise-robust speaker recognition. This section was designed to not only answer this question but also gain a deep understanding of the intrinsic noise robustness of GFCC and MFCC features. We first analyze all of their differences, which helps us to generate a number of assumptions. For each assumption, we design a corresponding set of experiments to test it. In this way, we are able to narrow down possible explanations, which eventually reveal the answer.

3.8.1 Differences in MFCC and GFCC Derivations

The HTK toolkit is frequently used to derive MFCC. Therefore, we take a closer look at the HTK version of the MFCC generation process.
MFCC Extraction (HTK version):

1. Pre-emphasize input signal
2. Perform short-time Fourier analysis to get magnitude spectrum
3. Wrap the magnitude spectrum into mel-spectrum using 26 triangular overlapping windows where center frequencies of the windows are equally distributed on the mel scale
4. Take the log operation on the power spectrum (i.e. square of mel-spectrum)
5. Apply DCT on the log-mel-power-spectrum to derive cepstral features
6. Perform cepstral liftering

The detailed process of GFCC extraction is listed as follows.

GFCC Extraction:

1. Pass input signal through a 64-channel gammatone filterbank
2. At each channel, fully rectify the filter response (i.e. take absolute value) and decimate it to 100 Hz as a way of time windowing. Then take absolute value afterwards. This creates a time-frequency (T-F) representation that is a variant of cochleagram
3. Take cubic root on the T-F representation
4. Apply DCT to derive cepstral features

Broadly speaking, there are two major differences. The obvious one is the frequency scale. GFCC, based on ERB scale, has finer resolution at low frequencies than MFCC (mel scale). The other one is the nonlinear rectification step prior to the DCT. MFCC uses a log while GFCC uses a cubic root. Both have been used in the literature. In addition, the log operation transforms convolution between excitation source and vocal
tract (filter) into addition in the spectral domain. Besides these two major differences, there are some other notable differences that are summarized in the following table. Next we analyze each of the differences in Table 3.4 in detail.

<table>
<thead>
<tr>
<th>Category</th>
<th>MFCC</th>
<th>GFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-emphasis</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td># of Frequency Bands</td>
<td>26</td>
<td>64</td>
</tr>
<tr>
<td>Cepstral Liftering</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Frequency Scale</td>
<td>Mel</td>
<td>ERB</td>
</tr>
<tr>
<td>Nonlinear Rectification</td>
<td>Logarithmic</td>
<td>Cubic Root</td>
</tr>
<tr>
<td>Scale-invariant (w/o 0th coefficient)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Intermediate T-F Representation</td>
<td>Mel Spectrum</td>
<td>Variant of Cochleagram</td>
</tr>
</tbody>
</table>

Table 3.4: Differences between MFCC and GFCC.

### 3.8.2 Diagnosis and Experimental Results

Our earlier results were reported on the 2002 NIST SRE dataset with 330 speakers. It is unclear if the noise robustness advantage of GFCC is specific to this dataset. To address this concern, we switch to the TIMIT corpus. Out of the entire 630 speakers, 330 are randomly chosen to match the number of speakers. Each speaker has 10 utterances and we choose 8 for training and remaining 2 for testing.

We scale clean training and test data to an average sound intensity (see (3.12) later) of 60 dB. Clean test data is then mixed with factory noise from the NOISEX-92 database (Varga and Steeneken, 1993) at an SNR of 0 dB. We use the IBM for voice activity
detection. In other words, the frames with at least one reliable T-F unit, labeled as 1 the IBM, are selected for recognition. Twenty-two-dimensional MFCC and GFCC, with 0\textsuperscript{th} coefficient removed, are used for this study. No cepstral mean normalization is performed as the long-term mean of cepstral features is not reliable to represent non-stationary noises such as factory noise. Speakers are modeled using GMM adapted from a 1024-component UBM. No speech separation is performed as the main goal is to evaluate the intrinsic noise robustness of MFCC and GFCC features.

First we establish the benchmark SID performance. The SID performance is shown in Table 3.5.

<table>
<thead>
<tr>
<th>Category</th>
<th>MFCC</th>
<th>GFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark performance</td>
<td>3.94</td>
<td>16.36</td>
</tr>
</tbody>
</table>

Table 3.5: Benchmark performance of GFCC and MFCC in terms of speaker recognition rates (%).

As shown in Table 3.5, the SID performance is rather poor in both cases due to a relatively low SNR and the absence of speech separation. However, it is obvious that GFCC is substantially more noise-robust than MFCC, which is consistent with the findings in Section 3.7.3.

\textit{Difference 1: Pre-emphasis}

One pre-processing difference between GFCC and MFCC is pre-emphasis. As a common practice, pre-emphasis is adopted in the HTK toolkit by applying a high-pass
filter with a setting of [1, -0.97]. High-pass filtering inevitably alters energy distribution across frequencies, as well as the overall energy level. This could have significant impact on the energy-related GFCC features. To verify if pre-emphasis degrades MFCC’s noise robustness, we remove it from MFCC and add it to GFCC. The resulting SID performance is shown in the following table.

<table>
<thead>
<tr>
<th>Category</th>
<th>MFCC</th>
<th>GFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/ Pre-emphasis</td>
<td>3.94 (default)</td>
<td>10.76</td>
</tr>
<tr>
<td>w/o Pre-emphasis</td>
<td>3.03</td>
<td>16.36 (default)</td>
</tr>
</tbody>
</table>

Table 3.6: Impact of pre-emphasis on GFCC and MFCC in terms of speaker recognition rates (%).

Table 3.6 suggests that removing pre-emphasis has little impact on MFCC’s noise robustness. Adding pre-emphasis to GFCC drops performance as expected, even though the altered GFCC is still substantially more robust. Therefore, pre-emphasis is not the answer.

**Difference 2: Number of Frequency Bands/Channels**

In the aforementioned MFCC generation, magnitude spectrum is wrapped into 26 mel-bands with triangular overlapping windows. GFCC uses the 64-channel gammatone filterbank as the front-end. It is reasonable to assume that more frequency channels may lead to better noise robustness. We test this assumption by increasing the number of
triangular windows and decreasing the number of channels of the gammatone filterbank. Table 3.7 presents the results.

<table>
<thead>
<tr>
<th>Category</th>
<th>MFCC</th>
<th>GFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>26 Bands/Channels</td>
<td>3.94 (default)</td>
<td>13.33</td>
</tr>
<tr>
<td>64 Bands/Channels</td>
<td>2.12</td>
<td>16.36 (default)</td>
</tr>
</tbody>
</table>

Table 3.7: Impact of number of frequency bands/channels on GFCC and MFCC in terms of speaker recognition rates (%).

As shown in Table 3.7, increasing the number of bands does not improve the robustness of MFCC. Using the 26-channel gammatone filterbank degrades the performance of GFCC by a small amount, but it is still substantially more robust than MFCC. Clearly, the number of frequency bands/channels is not the answer.

**Difference 3: Cepstral Lifting**

A post-processing difference between these two features is cepstral liftering. What it does is to filter cepstral coefficients. The “energy compaction” property of DCT makes higher dimensions of cepstral coefficients numerically very small. This is not a problem, but HTK toolkit purposely amplifies higher dimensional coefficients to balance the magnitudes and variances across dimensions for the sake of displaying parameters, variance flooring, etc. GFCC resolves the same problem by taking the lower 22-dimensional coefficients (without 0th coefficient). Therefore there is no need of cepstral liftering for GFCC. If no cepstral liftering is the reason of GFCC’s noise robustness, we
expect the robustness of MFCC will be substantially improved by dropping cepstral liftering. The experiments to test this assumption produce results shown in the following table.

<table>
<thead>
<tr>
<th>Category</th>
<th>MFCC</th>
<th>GFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/ Cepstral Liftering</td>
<td>3.94 (default)</td>
<td>16.36</td>
</tr>
<tr>
<td>w/o Cepstral Liftering</td>
<td>3.03</td>
<td>16.36 (default)</td>
</tr>
</tbody>
</table>

Table 3.8: Impact of cepstral liftering on GFCC and MFCC in terms of speaker recognition rates (%).

As illustrated in Table 3.8, there is no substantial performance change by adding or dropping cepstral liftering for either feature. Hence, cepstral liftering does not appear to be the reason.

**Difference 4: Frequency Scale**

We now evaluate the hypothesis of frequency scale. The argument is that the ERB scale has finer resolution than the mel scale at the low frequency range where speech energy primarily resides. However, recent studies on speaker recognition suggest that high frequency range also contains meaningful speaker information and should not be overlooked (Zhou et al., 2011). They also show that the linear scale is as robust as the mel scale in certain noisy conditions. This somewhat contradicts the hypothesis. To test this hypothesis, we make the triangular overlapping windows equally distributed on the ERB scale. In the meantime, we make sure the center frequencies of the gammatone
filters are equally distributed on the mel scale. If this hypothesis holds true, we expect a performance boost from MFCC and a performance drop for GFCC. Results are presented in Table 3.9.

<table>
<thead>
<tr>
<th>Category</th>
<th>MFCC</th>
<th>GFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mel Scale (26 bands)</td>
<td>3.94 (default)</td>
<td>17.88</td>
</tr>
<tr>
<td>ERB Scale (64 bands)</td>
<td>3.33</td>
<td>16.36 (default)</td>
</tr>
</tbody>
</table>

Table 3.9: Impact of frequency scale on GFCC and MFCC in terms of speaker recognition rates (%).

The results in Table 3.9 show that changing the frequency scale does not degrade GFCC’s robustness. At the same time, it does not improve MFCC’s robustness, either. Note that replacing ERB scale with mel scale even slightly improves the performance of GFCC. It appears that mel scale is a better scale. Hence, scale is not the answer.

**Difference 5: Log vs. Cubic Root and Scale Invariance**

Another main difference between the two features is the nonlinear rectification. It is directly correlated with the scale invariance property. Assuming an input signal $s(t)$ is scaled by a constant factor of $k$. Due to the linearity of Fourier analysis and triangular window wrapping, this constant factor is carried to the mel-spectrum. By taking the log operation on the mel-power-spectrum, the constant factor becomes an additive term. It is easy to prove that the DCT of a constant term is all 0 except for the $0^{th}$ coefficient. Therefore, by excluding $0^{th}$ coefficient (energy related), MFCC is the same no matter
how we scale the input signal. GFCC also carries the constant factor $k$ into the intermediate T-F representation because of linearity of gammatone filterbank. Nonetheless, the cubic root operation cannot convert the factor into an additive term. As DCT is linear, this cubic root of $k$ will be manifested in the cepstral coefficients. This is why GFCC is not scale-invariant. In other words, we do not get the same GFCC if we scale the input signal by a constant factor. This could make a difference in the noise robustness. We create new MFCC by replacing the log with the cubic root and similarly for GFCC by using the log. To be consistent, for the new MFCC, we drop pre-emphasis and cepstral liftering. We add them on the new GFCC. Results are shown in the following table.

<table>
<thead>
<tr>
<th>Category</th>
<th>MFCC</th>
<th>GFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>3.94 (default)</td>
<td>3.03</td>
</tr>
<tr>
<td>Cubic Root</td>
<td>28.48</td>
<td>16.36 (default)</td>
</tr>
</tbody>
</table>

Table 3.10: Impact of nonlinear rectification on GFCC and MFCC in terms of speaker recognition rates (%).

As demonstrated in Table 3.10, the noise robustness of MFCC is dramatically improved by taking the cubic root rectification. Meanwhile, GFCC loses its advantage by switching to the log. The nonlinear rectification is therefore a likely reason for GFCC’s superior robustness. It is worth pointing out that the new MFCC is even more robust than the regular GFCC, yielding 12% absolute improvement. With both features undergoing
the cubic root operation, the last difference would be the T-F representation prior to the DCT. Our results in Table 3.10 suggest that mel-power-spectrum is a better choice. We examine this in the next section.

3.8.3 Further Exploration

Study of Intermediate T-F Representation

MFCC comes from mel-power-spectrum where each element represents power/energy of the corresponding T-F unit. The T-F representation of GFCC is a variant of the cochleagram, which is a T-F representation with each element representing energy of the corresponding T-F unit. GFCC is derived by decimating filter responses instead of calculating energy at each T-F unit. The decimation process potentially throws away too much information. We now derive GFCC directly from the cochleagram. In other words, we apply the cubic root rectification on the cochleagram. We also explore different combinations of frequency scales and the number of frequency bands as shown in Table 3.11.
As seen in Table 3.11, GFCC derived from the cochleagram improves the SID performance to a comparable level with MFCC. Both obtain optimal performance when used together with the mel-scale and 26 bands/channels. This strongly indicates that the advantage of MFCC in Table 3.10 is mainly due to the intermediate T-F representation.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>GFCC-decimation</th>
<th>GFCC-cochleagram</th>
<th>MFCC-cubic root</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERB Scale (64 bands)</td>
<td>16.36 (default)</td>
<td>26.36</td>
<td>20.91</td>
</tr>
<tr>
<td>ERB Scale (26 bands)</td>
<td>13.33</td>
<td>23.18</td>
<td>26.36</td>
</tr>
<tr>
<td>Mel Scale (26 bands)</td>
<td>17.88</td>
<td>28.48</td>
<td>28.48</td>
</tr>
<tr>
<td>Mel Scale (64 bands)</td>
<td>15.91</td>
<td>20.91</td>
<td>24.55</td>
</tr>
</tbody>
</table>

Table 3.11: Impact of T-F representation on GFCC and MFCC in terms of speaker recognition rates (%).

As seen in Table 3.11, GFCC derived from the cochleagram improves the SID performance to a comparable level with MFCC. Both obtain optimal performance when used together with the mel-scale and 26 bands/channels. This strongly indicates that the advantage of MFCC in Table 3.10 is mainly due to the intermediate T-F representation.

Results on Other Noises and SNRs

All the previous analysis was made only in one noisy condition (an SNR of 0 dB and factory noise). We have performed the same experiments at an SNR of 6 dB with factory noise and get a similar performance profile. Experiments on two new noises, white noise and speech shape noise, further confirm the findings in this paper. All of these experiments have indicated that GFCC is more noise-robust than MFCC due to the nonlinear rectification. The new MFCC using the cubic root operation substantially improves the SID performance and even outperforms GFCC. Deriving GFCC from the cochleagram substantially improves its robustness and produces comparable or better results than the improved MFCC. The optimal number of bands/channels depends on
specific noisy conditions. Similar trends have also been observed using the 2002 NIST SRE dataset.

The Scale Variance Problem

As we pointed out in Section 3.8.2, GFCC is not a scale-invariant feature. To perform speaker recognition, we need to scale both training speech and clean test speech (not mixture) to a comparable energy level. We use an utterance level average sound intensity (in dB) as the measurement of overall energy level given in the following equation.

\[
E = 10 \log_{10} \left( \frac{\sum_{i} s^2(i)}{\text{length}(s(t))} \right)
\] (3.12)

where \(E\) is the average sound intensity and \(s(t)\) is the input signal. It is straightforward to scale training data to a desired intensity (e.g. 60 dB). However, it is not trivial to infer the intensity of the underlying target signal in a mixture and scale it to the same level of the training data. There are two ways to address this scale variance issue. One is to first estimate the input SNR. There are reasonably reliable SNR estimation algorithms in the literature. Given a roughly estimated SNR, we can readily infer the energy ratio between the target and interference and calculate the intensity of the target based on the intensity of the mixture. Another way is to estimate the IBM first. We then average the energy-related T-F representation only in reliable T-F units which are dominated by the target. This average can be used to normalize the entire T-F representation prior to the DCT.
operation. Training data can be processed similarly where all the T-F units are deemed reliable. Both techniques have been shown to be effective in our study.

3.9 Discussion

An important finding in our study is that GFCC features outperform conventional MFCC features under noisy conditions. We have conducted an in-depth study to understand this observation. Our experiments first confirm the superior robustness of GFCC relative to MFCC exists on a new corpus. By carefully examining all the differences between them, we conclude that nonlinear rectification mainly accounts for noise robustness differences. In particular, cubic root rectification provides more robustness to features than log rectification.

Why is the cubic root operation better? It might be the case that some speaker information is embodied through different energy levels. In a noisy mixture, there are target dominant T-F units or segments indicative of this energy information. The cubic root operation makes features scale-variant (i.e. energy level dependent) and helps to preserve this information. The log operation, on the other hand, does not encode this information. In addition, we have shown that by modifying MFCC extraction, substantial noise robustness improvement is also obtained. Since MFCC is widely used in automatic speaker and speech recognition, the findings of this study should shed new light on effective feature representations for noise robustness.

Shao and Wang (2008) used the SSC corpus, and achieved large performance gains (see also Li and Huang, 2010). However, we have found that such gains are somewhat
inflated by the large lexicon overlap between training and test material. The SSC corpus has a small vocabulary and a large amount of training data. Each sentence in SSC has a fixed grammar and every word appears in both training and testing data. This situation is similar to the TIDigits corpus discussed in Section 3.7.5. On the other hand, the NIST corpus is a standard dataset for speaker recognition, which is much closer to practical situations.

Shao and Wang (2008) also employed uncertainty decoding in conjunction with GF feature reconstruction. Theoretically, uncertainty decoding is expected to improve recognition performance as the contributions of unreliable feature dimensions are discounted during decoding. Our experiments show that ideal information about feature uncertainty can indeed bring about considerable performance improvement. However, with estimated uncertainty, the decoding process does not provide significant performance improvements due to inevitable errors in the estimation process.

In robust speech recognition, the reconstruction method shows better performance compared with bounded marginalization in larger vocabulary tasks (Raj et al., 2004; Srinivasan et al., 2006). In our SID results, marginalization generally produces better results than reconstruction. The effectiveness of marginalization for SID has been shown in a number of previous studies (El-Maliki and Drygajlo, 1999; Shao and Wang, 2006b; Pullella et al., 2008). We should note that speaker and speech recognition are two different tasks despite the fact that approaches are often shared between them.

Although the combined system in this chapter significantly outperforms the individual modules on the NIST dataset, the improvement on the TIDigits dataset is insignificant. The simple combination strategy in (3.11) seems to lose its advantage when
the performance profiles of the individual modules are similar. In such situations, more sophisticated methods of classifier combination may be needed.

We should point out that this chapter deals with additive noise in robust speaker identification, not handset/channel variations which are widely studied topics in robust speaker recognition. Are GF and GFCC features also robust to handset variations? There is no reason to believe so as these features are not designed for such variations. Whether common techniques for handling handset/channel variations such as CMN can be effectively combined with our approach to deal with both additive noise and channel distortions remains to be seen.
CHAPTER 4

ROBUST SPEAKER IDENTIFICATION IN NOISY AND REVERBERANT CONDITIONS

4.1 Introduction

In daily acoustic environments, additive noise, room reverberation and channel/handset variations conspire to pose considerable challenges to such systems. A lot of research has been devoted to dealing with individual challenges. However, efforts have rarely been made on the combined effects of noise and reverberation. In this chapter, we explore the combined effects of noise and reverberation in SID. We deal with reverberation by training models in noise-free reverberant conditions, while assuming little knowledge of the amount of reverberation in the test data. Meanwhile, noise is suppressed through a CASA approach that segregates speech by binary masking. We perform binary classification using a DNN. We utilize a CASA mask for SID in two ways, namely bounded marginalization and direct masking. The outputs of the two methods are combined to make the final SID decision.

The rest of the chapter is organized as follows. Section 4.2 gives an overview of the system and discusses front-end processing including DNN-based mask estimation.
Bounded marginalization and direct masking are introduced in Section 4.3, followed by evaluations in Section 4.4. We conclude this chapter in Section 4.5. This chapter has been published in Zhao et al. (2014a, 2014b).

4.2 System Overview and Front-end Processing

4.2.1 System Overview

Figure 4.1 shows the schematic diagram of the proposed system. Noisy speech is first passed through a DNN classifier to produce a binary T-F mask. Simultaneously we extract GF and GFCC. Each of the multiple training conditions produces one set of speaker models that is utilized independently. GF-based speaker models are fed to the bounded marginalization module, while GFCC-based speaker models to the direct masking module. Local decisions corresponding to different training conditions are first combined within each module and subsequently between two modules to make the final SID decision.
4.2.2 Mask Definitions in Noisy and Reverberant Conditions

In the definition of the IBM (see (3.3)), what constitutes the target signal is not a straightforward question in noisy and reverberant conditions. For example, the entire reverberant speech can be considered as the target and the reverberant noise as interference (Jin and Wang, 2009). We call this the Reverberant IBM (IBM_R). Meanwhile, one can choose only the early reflections of the reverberant speech as the target and everything else (i.e. late reverberation and reverberant noise) as interference (Roman and

Figure 4.1: Schematic diagram of the proposed speaker identification system
Woodruff, 2011). The resulting definition is named *Early-Reverberant IBM* (IBM$_{ER}$). If we treat only the direct path of the reverberant speech as the target, we can obtain *Direct-Sound IBM* (IBM$_{DS}$) (Mandel *et al.*, 2010). We explore these three IBMs in Section 4.4.

### 4.2.3 Mask Estimation via DNN

The definition of the IBM is based on the prior information of target and interference. In practice, we have to estimate the IBM. Recent work in CASA employs supervised classification for IBM estimation. GMMs (Kim *et al.*, 2009) and SVMs (Han and Wang, 2012) have been used in anechoic conditions. Motivated by their superior performance (Wang and Wang, 2013), we employ DNNs for mask estimation in this chapter. The employed mask estimation system is detailed below.

We use the standard generative-discriminative procedure to train DNNs. First, the DNNs are pretrained using restricted Boltzmann machines (RBMs) in an unsupervised and layerwise fashion. An RBM is a two-layer neural network with a visible layer $v$ and hidden layer $h$, and a stack of RBMs forms a very powerful generative model (Hinton *et al.*, 2006). The joint probability of an RBM is defined based on an energy function $E$:

$$p(v, h) = \frac{e^{-E(v, h)}}{Z}$$  \hspace{1cm} (4.1)

where $Z$ is a normalization term called partition function. More specifically, raw features are used to train the first RBM. We then take the hidden activations from the first RBM to train the second RBM, and so on. Since the inputs (raw features) are usually real
valued, we employ a Gaussian-Bernoulli RBM for the first layer and Bernoulli-Bernoulli RBMs for all subsequent layers. Assuming visible units are Gaussian random variables with unit variance, we can define the energy function $E$ for this Gaussian-Bernoulli RBM as:

$$E(v,h) = \frac{1}{2} \sum_i (v_i - a_i)^2 - \sum_j b_j h_j - \sum_{i,j} w_{ij} v_i h_j$$  \hspace{1cm} (4.2)$$

where $v_i$ and $h_j$ are the $i$th and $j$th unit of $v$ and $h$, $a_i$ and $b_j$ are the bias for $v_i$ and $h_j$, respectively, and $w_{ij}$ is the symmetric weight between $v_i$ and $h_j$.

Training an RBM requires maximizing the joint probability (4.2) with respect to network weights. Once pretrained, the weights from a stack of RBMs are used to initialize a standard feedforward network, which is then discriminatively fine-tuned using the backpropagation algorithm. Since our target labels are binary, we use the cross-entropy objective function for backpropagation:

$$E = \sum_m \left[ d^{(m)} \log(p^{(m)}) + (1 - d^{(m)}) \log(1 - p^{(m)}) \right]$$  \hspace{1cm} (4.3)$$

where $m$ indexes training samples, $d^{(m)}$ is the label of sample $m$ and $p^{(m)}$ is the corresponding network prediction (posterior probability).

Our separation system works as follows. We extract features from the cochleagram and train a subband classifier for each frequency channel to estimate the target-
dominance of each T-F unit, where the training labels are provided by the IBM. Since a decision needs to be made for each T-F unit, we extract unit level features from the subband signal within each T-F unit. In this chapter, we use the complementary feature set proposed in Wang et al. (2013), which consists of amplitude modulation spectrogram, RASTA-PLP, MFCC and pitch-based features. We used the DNNs described above as the subband classifiers.

4.3 Recognition Methodology and System Design

Over the past few decades, the GMM has been the predominant approach for speaker modeling (Reynolds, 1995). The GMM framework along with the UBM (Reynolds et al., 2000) is adopted for speaker modeling in this chapter. The feature space of a speaker is described as a linear combination of multivariate Gaussians that represent broad acoustic classes. Such Gaussians are usually parameterized with diagonal covariance matrices. Given speaker models, we employ different recognition methods by incorporating binary masking. At each frame, a binary mask divides the T-F units into a reliable group and an unreliable group. Multiple methods have been developed to deal with unreliable T-F units group such as marginalization, reconstruction, and direct masking. We use bounded marginalization and direct masking as two modules.

4.3.1 Bounded Marginalization Module

The basic idea of marginalization is to base recognition on reliable T-F units while removing the impact of unreliable ones. Conventional marginalization integrates over
unreliable T-F units in the entire range of feature values, e.g. minus infinity to positive infinity. Bounded marginalization sets realistic lower and upper bounds for the integration, which has proven beneficial in Chapter 3. Refer to Section 3.5 for more details.

4.3.2 Direct Masking Module

Direct masking is a recently proposed technique for coupling binary masking and speech recognition (Hartmann et al., 2013). In direct masking, one simply attenuates the noise-dominant T-F units using a constant gain, instead of estimating them as done in feature reconstruction. Cepstral features are then calculated directly from this masked representation or from the resynthesized target signal. Results have shown that this leads to competitive recognition performance compared to bounded marginalization and feature reconstruction. Therefore, we use direct masking in this chapter.

When the IBM is available, we retain target-dominant T-F units and attenuate noise-dominant T-F units by 26 dB. For estimated binary masks, we have found that using the outputs of the DNNs directly performs better than converting them to binary values. GFCC features for speaker recognition are extracted from the resynthesized target signal, which is obtained by applying the ratio mask (i.e. DNN output) to the mixture.
4.3.3 Reverberant Model Training

Speaker models trained in anechoic and noise-free conditions do not generalize well to reverberation. To characterize speaker feature distributions in such conditions, we train speaker models from reverberant environments.

Reverberation is usually characterized in terms of reverberation time \( T_{60} \), which describes the amount of time for the direct sound to decrease by 60 dB. Room reverberation is typically modeled as a convolution between a direct signal and an RIR which characterizes a specific reverberant condition.

To simplify the experimental settings while assuming little prior knowledge of testing reverberation, we simulate \( N \) reverberant environments covering a plausible range of \( T_{60} \). In this chapter, the range is chosen from 0s (anechoic condition) up to 1s, covering daily room environments (Kuttruff, 2000). These \( N \) reverberant conditions are chosen as the representatives of the range and expected to generalize to \( T_{60} \)’s between these representative values. We train a set of speaker models in each of the \( N \) conditions. Each set of speaker models characterizes a unique reverberant condition and is used independently for speaker recognition.

4.3.4 Multi-condition Fusion and Module Combination

In each of the two modules, SID decisions from the \( N \) sets of speaker models are first fused to generate module output. We then combine the outputs of the two modules to make the final SID decision. This design is elaborated below.
For an unknown test reverberant condition, each of the $N$ reverberant training conditions correlates with the test condition differently. The speaker models from the best matching conditions should be used. However, these correlations are unknown without ground truth information. Reverberation classification has been proposed to classify the test reverberant condition as one of the training conditions and select speaker models from the chosen condition for recognition (Peer et al., 2008; Akula et al., 2009). There are two problems with this idea. The first one occurs when the test condition does not match any of the training conditions. A hard classification is unlikely to work well. The second is that the idea was tested only in noise-free reverberant conditions. It is more challenging to perform such a classification task when background noise is present. Instead of reverberation classification, we propose to fuse the contributions from all training conditions. If done well, we expect that the best matching condition will dominate the fusion. If none of the training conditions match the test condition well, this fusion could leverage multiple contributions. As the score ranges from these conditions could be very different, we normalize before fusing them to make the final SID decision. Like (3.10), the normalization is described in the following equation,

$$\hat{s} = \frac{s - \min(s)}{\max(s) - \min(s)}$$

(4.4)

where $s$ refers to the output of a single training condition, which is a vector of scores with the number of elements equal to the number of speakers used in training. $\hat{s}$ denotes the
normalized score vector. We combine the $N$ normalized score vectors using a simple summation. We have also explored several other combination strategies and none of them significantly outperforms this simple summation. The fusion is performed in both bounded marginalization and direct masking modules.

The two modules address SID in noise from different perspectives. The bounded marginalization module works in the spectral domain and utilizes some information from unreliable T-F units. On the other hand, the direct masking attenuates unreliable T-F units uniformly and employs GFCC in the cepstral domain. GF and GFCC exhibit complementary properties for noise-robust SID in Chapter 3. We have observed that the errors of the two modules tend not to agree and the underlying speaker often achieves high scores in both modules. Hence, we combine these two modules to further improve SID performance. Similar to within-module fusion, we first apply score normalization (see (4.4)) and then simply add the module scores.

4.4 Evaluation and Comparison

4.4.1 Experimental Setup

We randomly drew 300 speakers from the 2008 NIST Speaker Recognition Evaluation dataset (short2 part of the training set). Each speaker has a telephone conversation excerpt of 5 minutes in total duration. We apply simple energy-based voice activity detection to remove the large chunks of silence in the excerpt. Then we divide the recording into 5s long pieces. Two pieces with the highest energy are selected as the test
data in order to provide sufficient speech information. The remaining pieces are used for training. On average there are about 20 training utterances per speaker. We employ the Matlab implementation of the image method of Allen and Berkley (1979) to simulate room reverberation (Habets, 2010); results with recorded impulse responses are given in Section 4.4.5. The range of $T_{60}$ is varied from 0 to 1s, which covers a broad range of realistic reverberant environments (Kuttruff, 2000). We simulate three rectangular rooms to obtain $3 T_{60}$’s: 300, 600 and 900 ms. For each $T_{60}$, we simulate 5 RIRs by randomly positioning a speech source and a receiver with the source-to-receiver distance fixed at 2m. Each training utterance is convolved with the 5 RIRs. Each speaker is modeled in these three $T_{60}$’s separately using the GMM-UBM framework (Reynolds et al., 2000). Test RIRs, on the other hand, are obtained from 7 simulated rooms corresponding to 7 $T_{60}$’s from 300 ms to 900 ms with the increment of 100 ms. Details of the simulated rooms are shown in Table 4.1. We simulate 3 pairs of RIRs in each room ($T_{60}$) by randomly positioning a speech source, a noise source and a receiver with both source-to-receiver distances fixed at 2m. The relative location of each source to the receiver determines an RIR. This results in 21 pairs of RIRs in total. Each test utterance is convolved with 2 pairs of RIRs that are randomly selected from the 21 pairs RIR library. Specifically, for each pair, the RIR corresponding to the speech source is used to convolve with the target speech and the other one with interference. Factory noise, SSN and destroyer engine room noise from the NOISEX-92 database are used as interference (Varga and Steeneken, 1993). We generate 5 SNRs for each noise from 0 to 24 dB with the increment of 6 dB. In total, each SNR of each noise has $300 \times 2 \times 2 = 1200$ test trials.
We use two-hidden-layer DNNs, which strike a balance between performance and computational overhead (Wang and Wang, 2013). We train DNNs separately for bounded marginalization and direct masking. The IBM is used to provide training labels. The IBM compares the local SNR of each T-F unit with the LC, which is typically set to 0 dB to indicate which source is stronger. Recent studies on speech intelligibility (Roman and Woodruff, 2011) and robust speech recognition (Narayanan and Wang, 2013), however, have shown that an LC of 0 dB is not always optimal, which is confirmed by our SID experiments. We will elaborate how LC is determined in the following subsection. We also compare the three reverberant IBM definitions in terms of SID performance. The DNN training set is created by mixing 50 utterances from 50 randomly selected speakers with the 3 noises, 3 training T₆₀’s (300, 600, and 900 ms), and 5 SNR conditions (−5, 0, 5, 10 and 15 dB). At each T₆₀, the 5 training RIRs are divided into two groups: three are used to create the DNN training set and the remaining two for a cross validation set. In

<table>
<thead>
<tr>
<th>T₆₀ (ms)</th>
<th>Training Rooms: length, width, height (m)</th>
<th>Testing Rooms: length, width, height (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>(5, 4, 3)</td>
<td>(5, 4, 3)</td>
</tr>
<tr>
<td>400</td>
<td>-</td>
<td>(6, 4, 3)</td>
</tr>
<tr>
<td>500</td>
<td>-</td>
<td>(7, 5, 4)</td>
</tr>
<tr>
<td>600</td>
<td>(7, 6, 4)</td>
<td>(7, 6, 4)</td>
</tr>
<tr>
<td>700</td>
<td>-</td>
<td>(8, 7, 5)</td>
</tr>
<tr>
<td>800</td>
<td>-</td>
<td>(8, 7, 6)</td>
</tr>
<tr>
<td>900</td>
<td>(9, 8, 7)</td>
<td>(9, 8, 7)</td>
</tr>
</tbody>
</table>

Table 4.1: Sizes of the simulated rectangular rooms
other words, the 50 DNN training utterances are convolved with 3 selected RIRs. The remaining 2 RIRs are used to distort another 10 randomly selected training utterances to create a validation set for DNN training. For each RIR, we randomly choose a noise source position within the same room to derive an RIR for the noise. This gives us noisy and reverberant DNN training and validation sets. Note that the RIRs used for speaker modeling as well as DNN training/testing do not overlap with the RIRs of SID evaluations. The DNNs are supervisedly fine-tuned using stochastic gradient descent with the objective function given in (4.3).

We extract 64-dimensional GF for bounded marginalization and 22-dimensional GFCC features for direct masking. We also extract 22-dimensional MFCC features for the sake of comparison. Speaker models are adapted from a 1024-component UBM that is trained by pooling training data from all the enrolled speakers (Reynolds et al., 2000). For each speaker, we train 3 sets of models in the three reverberant training conditions for GF, GFCC and MFCC respectively. In addition, we train a set of anechoic models for each feature to generate benchmark performance.

We perform SID in selected frames with some target information. We refer to the frames containing at least one reliable T-F unit as “active frames”. To balance the number of selected frames and the number of reliable T-F units per frame to qualify for selection, we use as the selection criterion the smaller of half of the number of all frequency channels (i.e. 32) and the median number of reliable T-F units of all active frames for a noisy and reverberant speech utterance. Given an active frame, it will be selected if its number of reliable units is greater than the criterion.
4.4.2 IBM Comparisons

Regarding different IBM definitions as discussed in Section 4.2.2, one important issue is which IBM is most effective for SID. A related issue is LC values in the IBM definition.

To address these issues, we set up a small experiment by randomly selecting 50 speakers from the TIMIT corpus. Each speaker has 10 utterances in total, 8 of which are used for training and the remaining 2 for testing. Training and testing data are mixed with newly simulated RIRs following the same procedure as described earlier except that the sampling frequency is 16000 Hz (8000 Hz for the NIST SRE dataset). Only factory noise is used in this experiment and the SNRs of the test set are −6, 0, 6, 12 and 18 dB. Ten RIRs are randomly selected from the 21 test RIR pairs, so there are 50*2*10 = 1000 test trials for each SNR condition.

Figure 4.2 gives an example of IBM_{DS}. We vary LC to generate different IBMs for the two modules separately. Results are averaged across the 5 SNRs. The figure indicates that an LC of −12 dB is the optimal choice for bounded marginalization. On the other hand, direct masking favors an LC of −18 dB. Note that the plateau of each plot is relatively wide and it shows that the proposed SID system is robust to LC choices. It is worth mentioning that the proposed multi-condition fusion idea works well as expected. We conduct similar experiments on the other two IBM definitions. Our obtained optimal LCs are listed in Table 4.2.
Figure 4.2: SID accuracy (%) using IBM$_{DS}$ with different LCs. BM denotes bounded marginalization, DM direct masking. (a) BM performance. (b) DM performance. Each point in the figure is averaged across all the test SNRs. Anechoic denotes speaker models trained in the anechoic condition. 300 ms, 600 ms and 900 ms denote speaker models trained in the corresponding T$_{60}$. Multi-condition Fusion combines the local decisions from the four sets of speaker models.
An example of the three IBM definitions is shown in Figure 4.3. The left plot in each panel was created with an LC of 0 dB. It retains a reasonable number of 1s in IBM\textsubscript{R}. However, it gets sparser for IBM\textsubscript{ER}, and IBM\textsubscript{DS} has very few 1s. This is expected as IBM\textsubscript{ER} and IBM\textsubscript{DS} treat part or all the reverberated speech as interference, and their effective SNRs are much lower than IBM\textsubscript{R}. As we choose the optimal LCs for bounded marginalization (the right plots), more 1s show up, and the three IBMs now exhibit similar patterns.

<table>
<thead>
<tr>
<th>IBM Definition</th>
<th>Bounded Marginalization</th>
<th>Direct Masking</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM\textsubscript{R}</td>
<td>(-4)</td>
<td>(-12)</td>
</tr>
<tr>
<td>IBM\textsubscript{ER}</td>
<td>(-4)</td>
<td>(-12)</td>
</tr>
<tr>
<td>IBM\textsubscript{DS}</td>
<td>(-12)</td>
<td>(-18)</td>
</tr>
</tbody>
</table>

Table 4.2: Optimal LCs (dB) for different IBM definitions
Figure 4.3: Illustrations of 3 IBM definitions on a TIMIT sentence with 0 dB SNR and 500 ms T₆₀. (a) IBMᵦ with two LCs, 0 dB on the left and −4 dB on the right. (b) IBMᵦ with two LCs, 0 dB on the left and −4 dB on the right. (c) IBMᵦ with two LCs, 0 dB on the left and −12 dB on the right. 1 is shown as white and 0 as black.
Next we explore the utilities of the three IBM definitions with their optimal LCs for SID. As shown in Table 4.3, the proposed system outperforms the individual modules for all three IBM definitions. IBM$_{DS}$ produces the best performance in all the categories. The other two IBM definitions achieve comparable performance. It remains to be seen if the performance advantage of IBM$_{DS}$ holds when estimated IBMs are employed and a larger dataset like the NIST SRE is used.

<table>
<thead>
<tr>
<th>IBM Definition</th>
<th>Bounded Marginalization</th>
<th>Direct Masking</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM$_R$</td>
<td>83.68</td>
<td>83.18</td>
<td>88.52</td>
</tr>
<tr>
<td>IBM$_{ER}$</td>
<td>84.48</td>
<td>83.82</td>
<td>88.86</td>
</tr>
<tr>
<td>IBM$_{DS}$</td>
<td>86.78</td>
<td>87.02</td>
<td>92.32</td>
</tr>
</tbody>
</table>

Table 4.3: SID accuracy (%) of the three IBM definitions. Performance is averaged across all the SNRs

### 4.4.3 Performance with Estimated IBM

We now establish benchmark SID performance of the NIST SRE dataset. We apply anechoic speaker models to the anechoic and all reverberant test sets where noise is excluded. As shown in Table 4.4, MFCC-based models achieve the best performance in both anechoic and reverberant conditions. When reverberation is included in the test set, the performance of all anechoic speaker models drops substantially due to the mismatch. After reverberation is included in speaker models, the performance is shown in Table 4.5. As shown in the table, the introduction of reverberation in the training data significantly
improves performance for all the features. The GF-based models even outperform MFCC-based models in some cases. Models trained in the $T_{60}$ of 600 ms achieves the best performance, probably because it lies in the middle of the test $T_{60}$ range.

<table>
<thead>
<tr>
<th>Features</th>
<th>Anechoic Test Set</th>
<th>Reverberant Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>97.83</td>
<td>77.08</td>
</tr>
<tr>
<td>GFCC</td>
<td>88.17</td>
<td>56.08</td>
</tr>
<tr>
<td>GF</td>
<td>95.00</td>
<td>54.42</td>
</tr>
</tbody>
</table>

Table 4.4: Benchmark SID performance (%) of reverberant speaker models in the reverberant test set. Note that the first column is the same as the last column of Table 4.4 as both use anechoic speaker models to recognize reverberant test data.

<table>
<thead>
<tr>
<th>Features</th>
<th>Anechoic Models</th>
<th>300 ms</th>
<th>600 ms</th>
<th>900 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>77.08</td>
<td>85.75</td>
<td>86.00</td>
<td>82.42</td>
</tr>
<tr>
<td>GFCC</td>
<td>56.08</td>
<td>75.17</td>
<td>77.33</td>
<td>73.92</td>
</tr>
<tr>
<td>GF</td>
<td>54.42</td>
<td>82.67</td>
<td>87.17</td>
<td>84.25</td>
</tr>
</tbody>
</table>

Table 4.5: Benchmark SID performance (%) of anechoic speaker models. Noise is excluded in the test set.

Now we evaluate the proposed system in the noisy and reverberant test set. We use GF for the bounded marginalization module. Both GFCC and MFCC are used in the direct masking module. When estimated IBMs are used, we notice that the inclusion of
anechoic speaker models in the multi-condition fusion stage does not help at all due to their substantial performance gap from reverberant speaker models. Therefore, we only fuse the reverberant speaker models in each module.

<table>
<thead>
<tr>
<th>Factory</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>24dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC_DM</td>
<td>12.50</td>
<td>23.00</td>
<td>43.50</td>
<td>64.83</td>
<td>75.67</td>
<td>43.90</td>
</tr>
<tr>
<td>GFCC_DM</td>
<td>33.25</td>
<td>48.33</td>
<td>61.00</td>
<td>70.33</td>
<td>74.00</td>
<td>57.38</td>
</tr>
<tr>
<td>GF_BM</td>
<td>34.08</td>
<td>49.17</td>
<td>59.25</td>
<td>71.92</td>
<td>80.00</td>
<td>58.88</td>
</tr>
<tr>
<td>Comb. Syst.</td>
<td>40.33</td>
<td>59.17</td>
<td>67.67</td>
<td>76.75</td>
<td>81.83</td>
<td>65.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destroyer</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>24dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC_DM</td>
<td>16.17</td>
<td>33.17</td>
<td>50.33</td>
<td>64.17</td>
<td>75.17</td>
<td>47.80</td>
</tr>
<tr>
<td>GFCC_DM</td>
<td>35.75</td>
<td>47.25</td>
<td>57.67</td>
<td>66.58</td>
<td>72.08</td>
<td>55.87</td>
</tr>
<tr>
<td>GF_BM</td>
<td>45.83</td>
<td>57.92</td>
<td>69.08</td>
<td>78.58</td>
<td>81.83</td>
<td>66.65</td>
</tr>
<tr>
<td>Comb. Syst.</td>
<td>50.00</td>
<td>61.67</td>
<td>73.50</td>
<td>79.75</td>
<td>82.42</td>
<td>69.47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSN</th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>24dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC_DM</td>
<td>18.50</td>
<td>34.33</td>
<td>55.92</td>
<td>71.33</td>
<td>79.58</td>
<td>51.93</td>
</tr>
<tr>
<td>GFCC_DM</td>
<td>37.67</td>
<td>52.92</td>
<td>64.33</td>
<td>70.92</td>
<td>74.92</td>
<td>60.15</td>
</tr>
<tr>
<td>GF_BM</td>
<td>44.83</td>
<td>61.17</td>
<td>73.58</td>
<td>79.83</td>
<td>83.42</td>
<td>68.57</td>
</tr>
<tr>
<td>Comb. Syst.</td>
<td>51.25</td>
<td>67.00</td>
<td>77.83</td>
<td>83.00</td>
<td>84.83</td>
<td>72.78</td>
</tr>
</tbody>
</table>

Table 4.6: SID accuracy (%) of the proposed system using IBM. MFCC_DM denotes the direct masking module with MFCC features. GFCC_DM denotes the direct masking module with GFCC features. GF_BM denotes the bounded marginalization module. Comb. Syst. denotes the proposed system.

Table 4.6 shows the SID performances of the proposed methods: the direct masking module with MFCC and GFCC, the bounded marginalization module with GF, and the
combined system. On average, the bounded marginalization module outperforms the direct masking module for both MFCC and GFCC. The direct masking module with GFCC substantially outperforms that of MFCC at the low SNRs, likely due to the better noise robustness of GFCC feature, as shown in Chapter 3. As the SNR increases, MFCC closes the gap and even outperforms GFCC. In the combined system, we employ the direct masking module with GFCC. The combined system outperforms individual modules in every test condition. For example, the combined system outperforms both modules by around 10% for factory noise at 6 dB.

<table>
<thead>
<tr>
<th></th>
<th>0dB</th>
<th>6dB</th>
<th>12dB</th>
<th>18dB</th>
<th>24dB</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Factory</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM_R</td>
<td>40.33</td>
<td>59.17</td>
<td>67.67</td>
<td>76.75</td>
<td>81.83</td>
<td>65.15</td>
</tr>
<tr>
<td>IBM_ER</td>
<td>37.92</td>
<td>56.83</td>
<td>67.67</td>
<td>74.75</td>
<td>79.00</td>
<td>63.23</td>
</tr>
<tr>
<td>IBM_DS</td>
<td>35.00</td>
<td>53.75</td>
<td>64.25</td>
<td>74.42</td>
<td>79.08</td>
<td>61.30</td>
</tr>
<tr>
<td><strong>Destroyer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM_R</td>
<td>50.00</td>
<td>61.67</td>
<td>73.50</td>
<td>79.75</td>
<td>82.42</td>
<td>69.47</td>
</tr>
<tr>
<td>IBM_ER</td>
<td>48.42</td>
<td>62.42</td>
<td>72.42</td>
<td>79.33</td>
<td>81.50</td>
<td>68.82</td>
</tr>
<tr>
<td>IBM_DS</td>
<td>43.00</td>
<td>57.08</td>
<td>70.17</td>
<td>77.83</td>
<td>80.50</td>
<td>65.72</td>
</tr>
<tr>
<td><strong>SSN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM_R</td>
<td>51.25</td>
<td>67.00</td>
<td>77.83</td>
<td>83.00</td>
<td>84.83</td>
<td>72.78</td>
</tr>
<tr>
<td>IBM_ER</td>
<td>49.67</td>
<td>66.75</td>
<td>75.67</td>
<td>81.42</td>
<td>81.58</td>
<td>71.02</td>
</tr>
<tr>
<td>IBM_DS</td>
<td>41.75</td>
<td>62.42</td>
<td>74.25</td>
<td>80.08</td>
<td>81.58</td>
<td>68.02</td>
</tr>
</tbody>
</table>

Table 4.7: SID performance (%) summary of the proposed system with different IBM definitions
Table 4.7 lists the SID results of the proposed system with the three IBM definitions. Compared to Table 4.3, the advantage of IBM_{DS} no longer exists. It is likely due to the quality of estimated masks. IBM_{DS} has very low LCs (−12 and −18 dB), which introduce substantial amounts of noise to the reliable T-F units, making it difficult for mask estimation. The performances of the other two IBMs definitions are still close. These results suggest that it is easier to estimate IBM_{R}.

4.4.4 Comparison with Related Systems

We pointed out that there was little study on the combined effects of reverberation and noise for SID. It is thus difficult to find comparison systems. As a result, we adapt a few related systems for the sake of comparison which should still provide useful perspectives on the relative performance of our model. The first related system, labeled as “Multi-conditional Training” in Figure 4.4, was designed for robust speaker verification using i-vector based techniques (Garcia-Romero et al., 2012). Each speaker is modeled as a GMM adapted from the UBM. A supervector is obtained by concatenating the means of Gaussians, and Garcia-Romero et al. map the supervector to a lower-dimensional i-vector. This system focuses on how to train from multiple training conditions followed by a combination to deal with noise and reverberation. A top performing scheme trains Gaussian probabilistic linear discriminant analysis models in both reverberant and noisy conditions. We implement this scheme for comparison due to its effectiveness and simplicity. In their experiments, multi-conditional training data were created by adding 3 types of noise: babble, car and helicopter, at 0 dB, 6 dB,
10 dB and 20 dB SNRs. Additional training data were produced by convolving clean speech with simulated RIRs at 100 ms, 300 ms and 500 ms reverberation times. Totally there are 16 training conditions, including the anechoic condition. In the implementation of this method, we use 19-dimensional MFCC features and their delta features to be consistent with the comparison system. Speaker models are trained by pooling training data from not only the reverberant training conditions we use, but also anechoic and noisy conditions (factory noise, destroyer engine room noise and SSN) in a wide range of SNRs (0, 6, 12, 18 and 24 dB). The second system, labeled as “Reverb. Classification”, was designed to deal with reverberation alone (Peer et al., 2008; Akula et al., 2009). It trains speaker models in multiple reverberant conditions separately. Given a test utterance, it first identifies the closest training condition and uses the models of that condition to perform speaker recognition, as detailed in Section 2.2 of Akula et al. (2009) and Section 3.4 of Peer et al. (2008). Since it only deals with noise-free reverberant speech, we apply our estimated CASA masks for noise suppression as front-end processing for this comparison system. More specifically, we use the UBMs trained separately in the 3 rooms to perform reverberation classification on noise-suppressed speech, which is consistent with Akula et al. and Peer et al. The third system, labeled as “Speech Enhancement”, uses a state-of-the-art speech enhancement algorithm to suppress noise (Hendriks et al., 2010). We use the source code of this algorithm from the authors to enhance the test speech. The last one, labeled as “Baseline”, directly recognizes the test data using MFCC-based anechoic speaker models.
Figure 4.4: SID comparisons of the proposed system with related systems under factory noise, destroyer engine room noise and speech shape noise using simulated RIRs.

continued
The performance comparison of all these systems along with the proposed system that employs IBM$_R$ is shown in Figure 4.4. The proposed system outperforms all the related systems in all the test conditions. The second best performing system is the reverberation classification method, which is partly due to the effectiveness of the supplied CASA masks. As a state-of-the-art system in noisy and reverberant conditions, the multi-conditional training method does a reasonable job at high SNRs, but not at low SNRs. Although the speech enhancement algorithm was proposed to deal with noisy speech in anechoic conditions, it exhibits reasonable performance in the reverberant environments, as shown by the large improvement over the MFCC baseline, particularly for the engine noise and SSN. It even outperforms multi-conditional training at low SNRs. It could be
the smearing effect of late reverberation on speech spectrum is somewhat similar to the corruption by SSN, which can be effectively attenuated by speech enhancement.

4.4.5 Evaluation with Real Impulse Responses

The results reported so far are generated using simulated RIRs. We now test our system using RIRs recorded in real rooms to assess its utilities in real environments. We use the RIRs collected in Bell Labs (Ward et al., 1994). There are three $T_{60}$’s (300 ms, 500 ms and 900 ms) and 4 RIRs are collected at each $T_{60}$ corresponding to 4 microphone positions. We observe that the actual $T_{60}$ values of these RIRs are probably much higher than the given values according to our measurements. For example, the RIRs of 900 ms would have a $T_{60}$ of 1.4 ~ 1.6 s as reported in (Gelbart, 2004). In this chapter, we use RIRs from the $T_{60}$’s of 300 ms and 500 ms. We use the third and fourth RIRs of each $T_{60}$ to create the training set of speaker models. To create the test set, we use the first RIR to convolve with speech and the second one with noise. We then switch these two to get another setting. Therefore we have 2 RIR pairs for each $T_{60}$. Each test utterance is randomly convolved with 1 of the 2 pairs at each $T_{60}$. The NIST dataset is employed and the remaining experimental setup stays the same as with simulated RIRs. Note that we use the DNNs trained on simulated RIRs for mask estimation. In other words, there is no retraining of DNNs using real RIRs.
Figure 4.5: Comparison of IBM$_R$ and estimated IBM$_R$ of an utterance mixed with SSN and real RIR ($T_{60} = 300$ ms) and SNR = 6 dB.

Figure 4.5 shows an example of the IBM estimation on real RIRs. As can be seen, the DNNs generalize reasonably well to real RIRs, which is encouraging. This is consistent with (Jin and Wang, 2011b), where an MLP-based mask estimation algorithm shows similar generalization results. As in Section 4.4.4, we compare the proposed system with four related systems over 3 noise types and performance is shown in Figure 4.6. The proposed system outperforms the related systems in all the test conditions. Compared to simulated RIRs (see Figure 4.4), the MFCC baseline is much worse. Even in the noise-free reverberant condition, the MFCC baseline only achieves 56.5% accuracy, which is around 20% lower than with the simulated RIRs. Similarly, the absolute performance of all systems including the proposed system all decreases. This indicates that the real acoustic environments are more challenging than simulated ones for speaker recognition. Overall, the proposed system and the related systems show similar performance trends in simulated and real reverberant environments.
Figure 4.6: SID comparisons of the proposed system with related systems under factory noise, destroyer engine room noise and speech shape noise using real RIRs. 

\textit{continued}
4.5 Discussion

The combined effects of noise and reverberation have been studied in human listeners (Nabelek and Pickett, 1974; Hazrati and Loizou, 2012), and the results indicate that they pose a greater challenge than individual effects. This study addresses the combined effects in the domain of robust speaker identification. Our benchmark performance suggests that reverberation alone poses a challenge for traditional SID systems already, as shown in the reduced performance of an MFCC baseline from 97.83% to 77.08% using simulated RIRs. Even in the least noisy situation (24 dB SNR), the combined effects further reduce the performance to 58.67% with SSN.
The multi-condition fusion idea investigated here alleviates the problem that it is difficult to accurately match training and testing reverberant conditions. We have observed that the best performing training condition tends to dominate the fusion results, rendering classification of testing reverberant conditions less significant. Module combination further leverages the complementary advantages of noise-robust SID approaches and features, consistent with results in Chapter 3. The noise susceptibility of MFCC makes it a poor choice for the direct masking module.

IBM estimation in noisy and reverberant condition is a very challenging task. Except for Jin and Wang (2011b), little research has been done on this topic. MLPs, SVMs and DNNs have shown promising results on IBM estimation in noisy conditions alone, and this chapter further considers reverberant conditions by using DNNs due to their performance.

As shown in Table 4.3, IBM_{DS} outperforms IBM_{R}. However, estimated IBMs in Table 4.7 outperform estimated IBM_{DS}. IBM_{R} retains T-F units with significant amounts of late reverberation that are detrimental to recognition due to its noise-like characteristics. On the other hand, IBM_{DS} is able to capture speech onsets that are relatively robust to reverberation. This may explain the advantage of IBM_{DS}. During mask estimation, the local SNRs of onset-related T-F units are quite low, as indicated by the choice of very low LCs. Therefore the derived features are unlikely discriminative to achieve good mask estimation performance. The T-F units mainly containing late reverberation in IBM_{R} also do not contain discriminative features. However, the resulting missing errors in IBM_{R} estimation would not be nearly as harmful as those for IBM_{DS} estimation. This could explain why estimated IBM_{R} yields better performance.
We have demonstrated the utilities of our system using RIRs recorded in real environments. It is encouraging to see that DNNs trained on simulated RIRs generalize well to real RIRs. However, we observe that speaker models trained using simulated RIRs perform less well with real RIRs, maybe because speaker models are built from frame level features which are distorted differently by simulated and real RIRs. On the other hand, mask estimation makes decisions based on energy comparisons, which are not much affected by the differences.

In this chapter, bounded marginalization is combined with direct masking rather than missing feature reconstruction for SID. This is because of better performance and simplicity of direct masking. A combination of the three does not offer further improvement. One observation from Chapter 3 and this chapter is that direct masking and reconstruction both underperforms bounded marginalization. This is somewhat contrary to robust speech recognition where direct masking and reconstruction outperform bounded marginalization in large vocabulary datasets (Raj et al., 2004; Srinivasan et al., 2006; Hartmann et al., 2013). One possible explanation is that our current number of speakers is relatively small. Another reason might be the fundamental differences of speech recognition and speaker recognition. Vocal tract shape, reflected in formant positions, determines a pronounced phoneme. The formant positions of different speakers are broadly similar for the same phoneme, but finer positional differences, due to different vocal tract sizes, may contribute to differentiating speakers. Additive noise distorts spectral envelope and make speakers more alike. The DCT transformation smears the noise across all the cepstral dimensions and may damage finer positional differences. This would not affect speech recognition as it is supposed to be insensitive to speaker
differences. The quality of spectral reconstruction appears not good enough to outperform bounded marginalization for SID. Direct masking in the spectral domain is similar to marginalization. Its inferior performance to bounded marginalization is likely due to the latter’s utilization of bounds from unreliable T-F units.
CHAPTER 5
COCHANNEL SPEAKER IDENTIFICATION IN
ANECHOIC AND REVERBERANT CONDITIONS

5.1 Introduction

State-of-the-art cochannel SID systems report nearly perfect performance on the SSC corpus (Hershey et al., 2010; Li et al., 2010). This corpus (Cooke and Lee, 2006), however, was tailored for robust speech recognition rather than speaker recognition. The relative small vocabulary and common words between training and testing reduce the difficulty of the SID task (see Section 3.9). In this chapter, we employ a NIST SRE dataset. We first explore two GMM-based methods: one jointly performs cochannel SID and separation (Shao and Wang, 2006a) and the other is a two-stage system (Guan and Liu, 2008; Li et al., 2010). The two methods are combined for further improvement. Then, we propose the first DNN-based cochannel SID system working in both anechoic and reverberant conditions. It trains a frame level multi-class DNN classifier that outputs the posterior probability of a frame being dominated by each speaker. Frame level decisions are integrated to make the final decision.
The rest of the chapter is organized as follows. Section 5.2 gives a system overview. We formulate the cochannel SID problem and introduce the proposed methods in Section 5.3, followed by evaluations in Section 5.4. We conclude this chapter in Section 5.5.

5.2 System Overview

Figure 5.1 shows the schematic diagrams of two proposed systems. The first is a GMM-based system that combines two cochannel SID methods. One method jointly conducts SID and separation. Specifically, we first hypothesize a pair of speakers. Then we search for the optimal assignment of speech segments given the speaker pair. The speaker pair with the highest likelihood is chosen as the output. The corresponding optimal assignment provides a solution to cochannel separation. The other method is a two-stage system that yields the state-of-the-art performance on the SSC corpus. The first stage creates a short list of most probable speaker candidates (e.g. top 10). The second stage combines the top speaker model with each of the rest and calculates a likelihood score for each combination at the frame level. Scores are integrated across frames and the output is the speaker combination with the best score. We then combine the SID scores from the two methods to get the final output. Details of the combination will be discussed in the next section.

The second proposed system is DNN-based. It trains a DNN using frame level features. The output layer has the same number of nodes as speakers. Only the two nodes corresponding to the underlying speakers have non-zero training labels. During testing, the frame level output is aggregated across time to generate the final output.
Figure 5.1: Schematic diagrams of two proposed cochannel speaker identification systems. (a). GMM-based system. (b). DNN-based system.
5.3 Problem Formulation and Identification Methodology

In this section, we formulate the cochannel SID problem and present the proposed systems.

5.3.1 Problem Formulation

Given an observation $O$, the goal of cochannel SID is to get the two underlying speakers $\hat{a}$ and $\hat{b}$ that generate the observation. This can be formulated as searching for the speaker pair with the highest posterior probability.

$$\hat{a}, \hat{b} = \arg \max_{\hat{a}, \hat{b}} P(\hat{a}, \hat{b} | O)$$

$$= \arg \max_{\hat{a}, \hat{b}} p(O | \hat{a}, \hat{b}) P(\hat{a}, \hat{b})$$

$$= \arg \max_{\hat{a}, \hat{b}} p(O | \hat{a}, \hat{b})$$

(5.1)

We apply the Bayes formula to convert the posterior probability to the likelihood of a joint distribution of two speakers, with the assumption that all speaker pairs are equally probable. $p(O)$ is not dependent on speakers and can thus be dropped from the calculation. The question now becomes how to calculate likelihoods of a joint distribution.

Shao and Wang (2003, 2006a) have introduced a variable $g$, to (5.1), to assign each speech segment to one of the two speaker sources. The derivation is shown as follows.
Here $X$ denotes a speech segment, $S$ the set of all segments, and $g$ an assignment vector of the same length as $S$. Each element of $g$ is a binary label that assigns the corresponding segment to a speaker. The number of assignments is exponential with respect the number of segments. The integration over all assignments is approximated as a $\max$ operation, assuming that the optimal assignment dominates the summation. By assuming that segments are independent, the problem reduces to finding the best assignment for each segment and the likelihood of the utterance is the multiplication of segment likelihoods. The speaker pair with the highest likelihood is the SID output. The corresponding optimal assignment also gives a solution to the cochannel separation problem by organizing segments into two groups. In other words, this method jointly performs cochannel SID and separation, so we name it joint SID & separation (JSS). We point out that the $\max$ operation reduces the time complexity from $O(K^2 \cdot 2^N)$ to $O(K^2 \cdot N)$ where $K$ is the number of speakers and $N$ is the number of segments.

Another way directly approximates the joint distribution. For example, one can use $\text{sum}$ or $\max$ of single speaker distributions to approximate the joint distribution.
\[
\hat{\lambda}_a, \hat{\lambda}_b = \arg \max_{\lambda_a, \lambda_b} p(O | \lambda_a, \lambda_b)
\]
\[
= \arg \max_{\lambda_a, \lambda_b} \left( \prod_{X \in S} p(X | \lambda_a, \lambda_b) \right)
\]
\[
\approx \arg \max_{\lambda_a, \lambda_b} \left( \prod_{X \in S} \frac{p(X | \lambda_a) + p(X | \lambda_b)}{2} \right) \tag{5.3}
\]

or

\[
\approx \arg \max_{\lambda_a, \lambda_b} \left( \prod_{X \in S} \max \{p(X | \lambda_a), p(X | \lambda_b)\} \right) \tag{5.4}
\]

As can be seen, the \textit{max} approximation of (5.4) is equivalent to the last step of (5.2).

We compare the performance of the two operations in Section 5.4.2.

Li \textit{et al.} have proposed a two-stage algorithm that produces state-of-the-art performance on the SSC corpus (Guan and Liu, 2008; Li \textit{et al.}, 2010). The first stage ranks speakers according to their posterior probabilities given the observation. The likelihood of each frame \(X\) given a speaker model \(\lambda\) is calculated as follows.

\[
p(X | \lambda) = \sum_{\text{gain}} \sum_k \pi_{\text{gain}} \sum_k \pi_k N(X | \mu_k + \text{gain}, \sigma_k^2) \tag{5.5}
\]

Here a variable, \textit{gain}, is introduced to represent energy/intensity levels. \(\pi_{\text{gain}}\) is the weight of a specific gain. Speakers are modeled as GMMs. \(\pi_k\), \(\mu_k\) and \(\sigma_k\) are the weight, mean and standard deviation of \(k\)th Gaussian, respectively. Log-spectral features are used.
as speaker features, so gains are equivalent to additive constants of the features, reflected in the Gaussian means. The posterior probability of each speaker given $X$ is calculated as follows.

$$p(\lambda | X) = \frac{p(X | \lambda)p(\lambda)}{\sum_m p(X | \lambda_m)p(\lambda_m)}$$

(5.6)

where $m$ is the speaker index. $P(\lambda)$ and $P(\lambda_m)$ are prior probabilities. Assuming that all the speakers are equally probable, the priors can be eliminated. Frame level posterior probabilities are aggregated across time to obtain utterance level probabilities.

$$\ell(\lambda) = \sum_t p(\lambda | X_t)$$

(5.7)

where $t$ is the frame index. Before the aggregation, a threshold (e.g. 0.9) is applied to retain the top frames.

Speakers are ranked based on the scores from (5.7). The top ten speakers are kept for the second stage where the top speaker is combined with each of the remaining nine. The weight of a composite Gaussian component is the product of the individual component weights. The means of individual components are compared and the mean and variance corresponding to the larger mean are kept as the mean and variance of the composite component. The composite GMMs are used for standard speaker recognition to get the best speaker pair. The time complexity is quadratic with respect the number of Gaussian
components and gains. Supposing $N$ Gaussian components and $G$ gain levels for each speaker, there would be $N^2 \cdot G^2$ composite components. A faster composition method is proposed to reduce the time complexity. For each speaker pair, the best gain and Gaussian component are identified first and treated as the base gain and component. The base component and gain are then combined with the other speaker’s components at different gain levels. In this way, the complexity is linear with respect to the number of components and gains. We will further discuss the computational complexity in Section 5.5. We point out that the composition operates on a per frame basis.

Li et al.’s two-stage algorithm is a fine-tuned version of Hershey et al.’s SID system. The first stage is almost the same with some differences on the Gaussian likelihood calculation and frame aggregation. Hershey et al.’s system keeps 6 most probable speakers from stage one and pairs the top speaker with each of the rest. In the second stage, Hershey et al.’s system uses a max-based EM algorithm to estimate the optimal gains for each speaker pair. The pair whose gain adapted models maximize the likelihood of the test utterance is selected as the output. Overall, the two systems yield the best performance on the SSC corpus with Li et al.’s average performance around 1% higher.

### 5.3.2 Combined Method

The methods discussed above solve the cochannel SID problem from different perspectives. JSS targets not only SID but also speech separation. Although it is logical to introduce an assignment variable, the hard assignment on segments may not work well for segments with large amounts of overlap. The direct approximation of the joint
distribution using *sum* or *max* might not satisfy the underlying distribution. On the other hand, Li *et al.* assign a probability to each speaker at a frame, which avoids a hard classification. It also takes different TIR scenarios into account via the gain variable. As for speaker features, JSS operates on cepstral features, while Li *et al.* work in the log-spectral domain. As observed in Chapter 3 and 4, cepstral features and spectral features may offer complementary advantages for SID. We therefore propose to combine these two methods.

There are many ways to combine the two methods. We have explored several ideas and the two best are shown in Table 5.1. The major difference is how the short list of 10 speakers is derived. The first method uses JSS to get the short list, while the second uses the first stage of Li *et al.* Subsequently the short lists are fed to the second stage of Li *et al.*, whose scores are combined with the JSS scores to make the final decision. The first method uses JSS to get the short list, while the second uses the first stage of Li *et al.* Subsequently the short lists are fed to the second stage of Li *et al.*, whose scores are combined with the JSS scores to make the final decision.
The aforementioned methods are GMM-based. In this section, we formulate cochannel SID as a discriminative learning problem, where we directly learn a mapping from cochannel observations to the corresponding speak identities. Specifically, we treat cochannel SID as a multi-class classification problem and employ DNN as the learning machine. To our knowledge, this is the first study of DNN-based cochannel SID.

<table>
<thead>
<tr>
<th>Name</th>
<th>Combination Method 1</th>
<th>Combination Method 2</th>
</tr>
</thead>
</table>
| Steps | 1. For every speaker pair, JSS finds an optimal assignment  
2. JSS calculates a score according to the assignment, and uses it as the score of the speaker pair  
3. The score of a speaker is defined as the maximum score among all pairs with the speaker  
4. Speakers are ranked by their scores. A short list of top 10 speakers is retained  
5. The short list is fed to the second stage of Li et al. and scores are calculated  
6. Scores of Steps 4 and 5 are combined to produce the final output  | 1. For every speaker pair, JSS finds an optimal assignment  
2. JSS calculates a score according to the assignment, and uses it as the score of the speaker pair  
3. The score of a speaker is defined as the maximum score among all pairs with the speaker  
4. The first stage of Li et al. produces a short list of top 10 speakers  
5. The second stage of Li et al. derives scores for the speaker pairs  
6. Retrieve scores of the top speakers from Step 3  
7. Scores of Steps 5 and 6 are combined to produce the final output |

Table 5.1: Illustration of the two proposed combination methods

5.3.3 DNN-based Cochannel SID

The aforementioned methods are GMM-based. In this section, we formulate cochannel SID as a discriminative learning problem, where we directly learn a mapping from cochannel observations to the corresponding speak identities. Specifically, we treat cochannel SID as a multi-class classification problem and employ DNN as the learning machine. To our knowledge, this is the first study of DNN-based cochannel SID.
We use frame level log-spectral features as input. To encode temporal context, we splice a window of 11 frames of features to train the DNN. The training target of the DNN is the true speaker identities. We use soft training labels where the two underlying speakers each have a probability of generating the current frame. The sum of their probabilities equals one, whereas the other speakers have zero probabilities. We compare frame level energy of two speakers and use their ratio for the soft labels. More specifically, we construct the IBM, and frame level energy of each speaker is readily calculated from the mixture cochleagram according to the IBM.

The DNN employed in our study is a deep multi-layer perceptron. The DNN uses three hidden layers, each having 1024 sigmoidal hidden units. The standard backpropagation algorithm coupled with dropout regularization (dropout rate 0.2) is used to train the network. No unsupervised pretraining is used, as we have sufficient labeled data. We use the adaptive gradient descent along with a momentum term as the optimization technique. A momentum rate of 0.5 is used for the first 5 epochs, after which the rate increases to 0.9. We use a softmax output layer and cross-entropy as the loss function.

5.3.4 Model Training

In this chapter, we deal with both anechoic and reverberant test conditions. For the anechoic condition, we use anechoic data to train GMMs and DNNs. However, such models do not generalize well to reverberant conditions. Thus, we directly model speakers in the reverberant environments.
Assuming no knowledge of test reverberant conditions, we simulate \( N \) representative reverberant training conditions covering a plausible range of \( T_{60} \). Chapter 4 has shown that this technique has reasonable generalization. We prepare training data in each of the \( N \) conditions. GMMs are trained using single speaker data while DNNs are trained with cochannel data mixed at different TIRs. Details are given in the next section.

### 5.4 Evaluation and Comparison

#### 5.4.1 Experimental Setup

We randomly select 100 speakers from the 2008 NIST SRE dataset (\textit{short2} part of the training set). The telephone conversation excerpt of each speaker is roughly 5 minutes long. Large chunks of silence in the excerpt are removed. Then we divide the recording into 5 s pieces. Two pieces with the highest energy are used for tests in order to provide sufficient speech information. The rest is used for training. Overall each speaker has about 20 training utterances. Refer to Section 4.4.1 for more details.

Same as Chapter 4, Habets’s (2010) Matlab implementation of the image method is used to simulate room reverberation. We focus on the \( T_{60} \) range up to 1 s that covers realistic reverberant conditions. Three rooms are simulated to obtain 3 training \( T_{60} \)’s: 300, 600 and 900 ms. For each \( T_{60} \), we generate 5 RIRs by randomly positioning the source and receiver while keeping their distance fixed at 2 m. Each training utterance is convolved with the 5 RIRs of each room to create reverberant training data. Seven rooms are simulated to obtain 7 test \( T_{60} \)’s from 300 ms to 900 ms with a step size of 100 ms. We
randomly generate 3 pairs of RIRs at each T\textsubscript{60} where each pair provides one RIR for the target and one for the interferer. In total there are 21 pairs of test RIRs. Note that the RIRs are different between training and testing even when they are generated with the same T\textsubscript{60}.

For JSS, we extract 22-dimensional MFCC as speaker features. Speaker models are adapted from a 1024-component UBM trained by pooling training data from all the speakers. For Li et al., we extract 64-dimensional log-spectral features for GMM training. Specifically, a 64-channel gammatone filterbank is employed as the front-end. The filter output is converted to the cochleagram. We take the log operation on the cochleagram to get the features. For anechoic conditions, a 256-component GMM is trained for each speaker. Another 256-component GMM is trained using the reverberant training data by convolving the anechoic training data with the RIRs at 3 T\textsubscript{60}’s.

DNNs are trained using cochannel training data. Instead of one DNN per speaker, we train a universal DNN for all the speakers. We include training data from every speaker pair for a complete coverage. For anechoic conditions, we create 10 anechoic cochannel utterances per speaker pair at 3 TIRs (−5, 0 and 5 dB). In total, there are 4950 speaker pairs and 49500 cochannel training utterances per TIR. For reverberant conditions, we create 10 reverberant cochannel utterances at each of the 3 T\textsubscript{60}’s and 3 TIRs. In total, there are 49500 cochannel training utterances per TIR and per T\textsubscript{60}.

Cochannel test set covers all possible speaker pairs. For each pair, we create two anechoic utterances and two reverberant utterances at −5, 0 and 5 dB TIRs. There are totally 9900 anechoic test utterances and 9900 reverberant test utterances per TIR. Each
reverberant cochannel test utterance is created using a randomly selected RIR pair from the 21 RIR pair library.

5.4.2 Frame Selection for JSS and Max vs. Sum

Shao and Wang (2006a) employed a multi-pitch tracking algorithm to identify frames with only one pitch point. JSS operates on such frames. The rationale is that the single pitch frames should contain voiced speech from one speaker, and either unvoiced speech or nothing from the other speaker. Usually voiced speech has stronger energy, and is more characteristic of speaker identity. However, such a hard decision ignores unvoiced speech and overlapping voiced speech, which could be helpful for cochannel SID. We conduct the following experiments to investigate whether overlapping voiced speech and unvoiced speech are helpful for cochannel SID.

Evaluations are performed on the SSC corpus. We use Praat (Boersma and Weenink, 2007) to extract ground truth pitch from the premixed signals. We apply JSS to different types of frames and treat each frame as a segment. The results are shown in Table 5.2, where SID accuracy is measured as the percent of cochannel utterances where both speakers are correctly identified.
The average number of frames per utterance is 192 for the SSC corpus. Out of them, 75 are 1 pitch frames while 117 have either 2 pitch points or none. The 0 pitch frames correspond to unvoiced speech or silence, and the performance in such frames is around 60%, which is the worst. The 2 pitch frames yield slightly better performance than the 1 pitch frames, probably because the max approximation models voiced+voiced speech better than voiced+unvoiced speech or single voiced speech frames and there are more 2 pitch frames (6 on average). One important observation is that the combination of 0 pitch and 2 pitch frames further lift the performance to 91%. The combination of all types of frames yields the best performance.

The above results are generated using the max operation. We also run the same experiments using sum operation. The performance profile is very similar, and the best performance is obtained by combining all types of frames, shown in the last row of Table 5.2. Clearly the max approximation gives much better results, and therefore we use the max operation and perform SID on all the frames in the following sections.

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>Avg. Number</th>
<th>−9 dB</th>
<th>−6 dB</th>
<th>−3 dB</th>
<th>0 dB</th>
<th>3 dB</th>
<th>6 dB</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 pitch frames</td>
<td>75</td>
<td>80.17</td>
<td>81.67</td>
<td>85.67</td>
<td>85.50</td>
<td>85.00</td>
<td>84.17</td>
<td>83.70</td>
</tr>
<tr>
<td>0 pitch frames</td>
<td>36</td>
<td>58.17</td>
<td>58.00</td>
<td>58.67</td>
<td>61.33</td>
<td>60.50</td>
<td>59.83</td>
<td>59.42</td>
</tr>
<tr>
<td>2 pitch frames</td>
<td>81</td>
<td>78.17</td>
<td>85.00</td>
<td>89.50</td>
<td>91.17</td>
<td>89.50</td>
<td>84.00</td>
<td>86.22</td>
</tr>
<tr>
<td>0 or 2 pitch frames</td>
<td>117</td>
<td>83.67</td>
<td>89.83</td>
<td>93.33</td>
<td>95.00</td>
<td>94.83</td>
<td>91.67</td>
<td>91.39</td>
</tr>
<tr>
<td>All frames</td>
<td>192</td>
<td>89.83</td>
<td>95.33</td>
<td>98.00</td>
<td>98.50</td>
<td>98.17</td>
<td>96.83</td>
<td>96.11</td>
</tr>
<tr>
<td>All frames (sum)</td>
<td>192</td>
<td>67.83</td>
<td>75.5</td>
<td>84.33</td>
<td>88.17</td>
<td>81.83</td>
<td>75.67</td>
<td>78.89</td>
</tr>
</tbody>
</table>

Table 5.2: SID accuracy (%) of JSS in different types of frames. Max operation is used to approximate the joint distribution except for the last row (sum operation).
5.4.3 Performance on SSC Corpus

The state-of-the-art cochannel SID systems of Hershey et al. and Li et al. have reported performance on the SSC corpus. This corpus consists of 17000 training utterances from 34 speakers. Each training utterance is created following a fixed grammar: command, color, preposition, letter, number, and adverb. Each of the six positions has a small number of word choices. The cochannel test set comprises six TIRs from -9 dB to 6 dB. There are 600 test utterances for each TIR, and the test utterances follow the same grammar and share the same vocabulary as the training utterances.

We evaluate on this dataset first in order to make a direct comparison. Table 5.3 gives the SID results. As can be seen, our implementation of Li et al.’s two-stage system achieves the same average performance as in their paper. Both combination methods produce comparable performance to the state-of-the-art methods. The first method is slightly worse than Li et al. This is likely because the short list from JSS is not as reliable as that from Li et al., as indicated by their respective performances (96.1% vs. 99.0%). The DNN-based system yields the best results, although the performance gain is probably not statistically significant. Since the results are nearly perfect, there is not much room to improve and we can conclude that the proposed systems work comparably well.
As mentioned earlier, the nearly perfect SID performance might be caused by the easiness of the SSC corpus for cochannel SID. We now turn to the NIST SRE dataset.

### 5.4.4 Performance on NIST SRE Dataset with 50 Speakers

First we test on a subset of 50 speakers with 1225 speaker pairs, to be roughly comparable with the SSC corpus in terms of speaker number. We create two cochannel utterances for each pair at each of 3 TIRs, −5 dB, 0 dB and 5 dB. In total, there are 2450 test trials per TIR. The performance is given in Table 5.4. As shown in the table, there is a substantial drop of performance compared to the SSC corpus, confirming that the SSC corpus is rather easy for cochannel SID evaluation. For this dataset, JSS outperforms Li *et al.* by an average of 4.3%. The proposed combination methods significantly outperform the individual methods. We also evaluate the DNN-based cochannel SID system, which outperforms the better combination performance by almost 9%.

<table>
<thead>
<tr>
<th>Method</th>
<th>−9 dB</th>
<th>−6 dB</th>
<th>−3 dB</th>
<th>0 dB</th>
<th>3 dB</th>
<th>6 dB</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reported Performance of Hershey <em>et al.</em></td>
<td>96.5</td>
<td>98.1</td>
<td>98.2</td>
<td>99.0</td>
<td>99.1</td>
<td>98.4</td>
<td>98.2</td>
</tr>
<tr>
<td>Reported Performance of Li <em>et al.</em></td>
<td>97.3</td>
<td>98.8</td>
<td>99.5</td>
<td>99.7</td>
<td>99.7</td>
<td>98.8</td>
<td>99.0</td>
</tr>
<tr>
<td>JSS</td>
<td>89.8</td>
<td>95.3</td>
<td>98.0</td>
<td>98.5</td>
<td>98.2</td>
<td>96.8</td>
<td>96.1</td>
</tr>
<tr>
<td>Li <em>et al.</em></td>
<td>96.7</td>
<td>99.0</td>
<td>99.5</td>
<td>99.7</td>
<td>100.0</td>
<td>99.2</td>
<td>99.0</td>
</tr>
<tr>
<td>GMM Comb. Method 1</td>
<td>96.7</td>
<td>99.0</td>
<td>99.3</td>
<td>99.3</td>
<td>99.5</td>
<td>99.0</td>
<td>98.8</td>
</tr>
<tr>
<td>GMM Comb. Method 2</td>
<td>96.7</td>
<td>99.0</td>
<td>99.7</td>
<td>99.7</td>
<td>100.0</td>
<td>99.5</td>
<td>99.1</td>
</tr>
<tr>
<td>DNN</td>
<td>98.3</td>
<td>99.5</td>
<td>100</td>
<td>99.8</td>
<td>100</td>
<td>99.0</td>
<td>99.4</td>
</tr>
</tbody>
</table>

Table 5.3: SID accuracy (%) on SSC Corpus

As mentioned earlier, the nearly perfect SID performance might be caused by the easiness of the SSC corpus for cochannel SID. We now turn to the NIST SRE dataset.
Next we test in the reverberant conditions, and the results are shown in Table 5.5. As can be seen, the performances of all the methods degrade in the reverberant conditions. JSS drops by about 30%. Li et al.’s is slightly more robust, but still drops by more than 20%. Like in Table 5.4, both combination methods outperform the state-of-the-art performance. In addition, the proposed DNN-based system continues to perform the best, outperforming the better combination by more than 11%.

Table 5.4: SID accuracy (%) on NIST SRE dataset with 50 speakers

<table>
<thead>
<tr>
<th>Method</th>
<th>−5 dB</th>
<th>0 dB</th>
<th>5 dB</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSS</td>
<td>82.24</td>
<td>83.51</td>
<td>80.12</td>
<td>81.96</td>
</tr>
<tr>
<td>Li et al.</td>
<td>77.02</td>
<td>79.96</td>
<td>75.84</td>
<td>77.61</td>
</tr>
<tr>
<td>GMM Combination Method 1</td>
<td>86.41</td>
<td>86.69</td>
<td>84.20</td>
<td>85.77</td>
</tr>
<tr>
<td>GMM Combination Method 2</td>
<td>85.63</td>
<td>86.16</td>
<td>82.69</td>
<td>84.83</td>
</tr>
<tr>
<td>DNN</td>
<td>94.12</td>
<td>96.90</td>
<td>92.69</td>
<td>94.57</td>
</tr>
</tbody>
</table>

Next we test in the reverberant conditions, and the results are shown in Table 5.5. As can be seen, the performances of all the methods degrade in the reverberant conditions. JSS drops by about 30%. Li et al.’s is slightly more robust, but still drops by more than 20%. Like in Table 5.4, both combination methods outperform the state-of-the-art performance. In addition, the proposed DNN-based system continues to perform the best, outperforming the better combination by more than 11%.
5.4.5 Performance on NIST SRE Dataset with 100 Speakers

The SID task becomes more challenging as the number of speakers (classes) increases. To quantify cochannel SID dependency on number of speakers, we have performed cochannel SID evaluation by increasing the number of speakers from 50 to 100, quadrupling the number of classes to 4950. The SID results shown in Table 5.6 demonstrate that the combination methods outperform the individual ones, albeit by a smaller extent. As in the previous results, the default DNNs with has 3 hidden layers with 1024 nodes each outperform the best combination. With the increase of speaker size as well as training data size, we have also explored a few different DNN configurations. As we increase the number of units from 1024 to 2048 for each hidden layer, the SID performance improves by around 4.5%. There is a slight improvement as we expand the number of hidden layers from 3 to 5 without changing the hidden layer size, for either 1024 or 2048 hidden units. Further enlargement of the DNN size is expected to improve the performance even more, but at the expense of substantially increased computational complexity.

<table>
<thead>
<tr>
<th>Method</th>
<th>−5 dB</th>
<th>0 dB</th>
<th>5 dB</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSS</td>
<td>51.43</td>
<td>53.76</td>
<td>49.51</td>
<td>51.57</td>
</tr>
<tr>
<td>Li et al.</td>
<td>55.02</td>
<td>59.35</td>
<td>56.37</td>
<td>56.91</td>
</tr>
<tr>
<td>GMM Combination Method 1</td>
<td>57.84</td>
<td>60.20</td>
<td>56.73</td>
<td>58.26</td>
</tr>
<tr>
<td>GMM Combination Method 2</td>
<td>58.73</td>
<td>62.16</td>
<td>58.37</td>
<td>59.75</td>
</tr>
<tr>
<td>DNN</td>
<td>70.86</td>
<td>75.31</td>
<td>66.29</td>
<td>70.82</td>
</tr>
</tbody>
</table>

Table 5.5: SID accuracy (%) on reverberant NIST SRE dataset with 50 speakers
5.4.6 Scalability Study of GMM-based and DNN-based Approaches on Cochannel SID

The previous subsection indicates that there is a substantial performance drop as the number of speakers goes up. This is expected as SID is more prone to error with more speaker models to choose from. An interesting question is whether GMM and DNN based approaches show different scalability to speaker set size. In addition, does reverberation impact scalability? The following experiments are conducted to address these issues.

Li et al. and the default DNN configuration (i.e. 3 hidden layers with 1024 units each) are employed to represent GMM-based and DNN-based approaches respectively. We choose an anechoic test condition and the reverberant test conditions with T$_{60}$ of 600 ms.

<table>
<thead>
<tr>
<th>Method</th>
<th>-5 dB</th>
<th>0 dB</th>
<th>5 dB</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSS</td>
<td>39.59</td>
<td>41.76</td>
<td>38.70</td>
<td>40.02</td>
</tr>
<tr>
<td>Li et al.</td>
<td>43.58</td>
<td>47.12</td>
<td>43.58</td>
<td>44.76</td>
</tr>
<tr>
<td>GMM Combination Method 1</td>
<td>46.34</td>
<td>49.18</td>
<td>45.32</td>
<td>46.95</td>
</tr>
<tr>
<td>GMM Combination Method 2</td>
<td>44.55</td>
<td>47.84</td>
<td>44.53</td>
<td>45.64</td>
</tr>
<tr>
<td>DNN (1024 by 3)</td>
<td>52.67</td>
<td>59.78</td>
<td>52.58</td>
<td>55.01</td>
</tr>
<tr>
<td>DNN (1024 by 5)</td>
<td>54.13</td>
<td>61.31</td>
<td>54.33</td>
<td>56.59</td>
</tr>
<tr>
<td>DNN (2048 by 3)</td>
<td>56.91</td>
<td>64.76</td>
<td>56.99</td>
<td>59.55</td>
</tr>
<tr>
<td>DNN (2048 by 5)</td>
<td>57.32</td>
<td>64.82</td>
<td>57.52</td>
<td>59.89</td>
</tr>
</tbody>
</table>

Table 5.6: SID accuracy (%) on reverberant NIST SRE dataset with 100 speakers
We systematically increase the number of speakers from 10 to 100 and make sure the only varying variable is the number of speakers. The resulting performance is shown in Figure 5.2. There are a number of observations from Figure 5.2. GMM and DNN-based approaches both work very well with the small speaker set of 10, even in the reverberant conditions. Both approaches show a decline of performance with the increase of speaker set size. Reverberation exacerbates the degradation. Overall, the DNN-based approach declines at a much slower pace than the GMM-based approach in the anechoic condition, indicating better scalability to speaker set size. However, none of them scale well in the reverberant conditions, although the DNN-based approach holds a sizeable advantage.

Figure 5.2: Scalability study of DNN and GMM-based approaches with respect to the number of speakers.
5.5 Discussion

Cochannel SID is an important problem with real applications. Previous studies approach this problem from different perspectives such as the utilization of usable speech, and joint SID and cochannel separation. State-of-the-art methods achieve almost perfect performance on the SSC corpus. This study investigates whether these methods work on a standard speaker recognition corpus. The results suggest that the problem gets considerably more difficult on the 2008 NIST SRE dataset, as illustrated by a performance drop of more than 20% with Li et al.’s system.

Usable speech based methods usually ignore the overlapping speech and focus on homogenous speech segments. Our study demonstrates that “non-usable” speech is also helpful for cochannel SID. The joint speaker distributions are often approximated by some combination of individual speaker distributions. The difficulty of directly modeling the interaction lies in computational complexity, as pointed out by Hershey et al. For $K$ speakers and $G$ gain conditions, a complete coverage includes $O(K^2 \cdot G^2)$ speaker and gain combinations. Assuming a $C$ component GMM for each combination, each test frame requires $O(C \cdot K^2 \cdot G^2)$ Gaussian likelihood computations. Li et al. greatly reduces the complexity using individual speaker models. Its first stage requires $O(C \cdot K \cdot G)$ Gaussian likelihood computations to derive a short list of top 10 speakers. With the second stage working with a constant number of speaker models, the total computational complexity of Li et al. is $O(C \cdot K \cdot G)$. On the other hand, DNN trains a single neural network for all speakers. For a $M$ hidden layer network with $N$ units each, the computational complexity is $O((D+K)N + (M-1)N^2)$, where $(M-1) \cdot N^2$ denotes the computations among the hidden
layers, and $D \cdot N$ and $K \cdot N$ the input ($D$ indicates feature dimensionality) and the output layer, respectively. By treating $C$, $G$, $D$, $M$, and $N$ as predetermined constants, the time complexities of Li et al. and our proposed DNN system are both $O(K)$, in other words, linear with respect to the number of speakers.

Scalability is a concern for real applications as the number of speakers may not be small. The performance is expected to degrade because the number of speaker pairs increases quadratically. Our study shows that the DNN-based approach maintains good performance as speaker set size grows from 10 to 100 in the anechoic condition. However, scalability becomes an issue for both DNN and GMM-based approaches in reverberant conditions. One possible explanation is that the smearing effects of reverberation make speaker features (such as pitch) more alike and thus reduce the discriminability of the GMM models and the DNN classifiers.

We have also explored hard training labels for the DNN. Specifically, the two underlying speakers have a label of 1 and everyone else 0. In order to train the DNN with the hard labels, we use sigmoidal output units and explore loss functions of mean squared errors and cross-entropy. The two functions produce similar performance that is significantly better than the combination methods but consistently worse than using soft training labels.
CHAPTER 6

CONTRIBUTIONS AND FUTURE WORK

6.1 Contributions

This dissertation addresses robust SID in three aspects. First, we study noise-robustness of an SID system. Then we incorporate a new training strategy and derive an SID system that is robust to the combined effects of noise and reverberation. Subsequently we propose a GMM-based cochannel SID system that combines two existing methods. Lastly, we make the first attempt of employing DNN for cochannel SID.

Chapter 3 proposes a noise-robust SID system that combines two missing feature methods. The first contribution is that we establish an effective combination. In the combination, we leverage the complementary advantages of spectral and cepstral features, feature reconstruction and bounded marginalization. Secondly, we incorporate MLP-based speech separation to identify reliable and unreliable T-F units. Channel-specific MLPs are able to learn a mapping from T-F unit level features to a posterior probability of the T-F unit being speech dominant. Our third contribution is our
demonstration and analysis of GFCC’s superior noise-robustness compared to other speaker features.

In Chapter 4, we develop a monaural robust SID system that combats the combined effects of noise and reverberation. Our first contribution is a study on how the combined effects impact SID performance. Several similar studies employ more than one microphone. Others focus either on reverberation or noise. The second contribution lies in our combination of bounded marginalization and direct masking. Direct masking is effective approach in robust speech recognition. Our study is the first to apply it to robust SID. Our results indicate that it is complementary to bounded marginalization in noisy and reverberant conditions. The next contribution is our systematic evaluations on different reverberant IBM definitions. Three IBM definitions are examined. We compare the difficulty of their estimation and respective SID performance. We find that IBM_{DS} achieves the best SID performance. However, IBM_{R} outperforms both IBM_{R} and IBM_{DS} when estimated IBMs are used. The last contribution is our incorporation of a DNN-based classifier for mask estimation.

We address the cochannel SID problem in Chapter 5. The primary contribution is the proposal of a DNN-based cochannel SID system. We formulate cochannel SID as a multi-class classification task. This is the first attempt of applying DNN to cochannel SID. In addition, we evaluate two existing GMM-based methods and exploit their complementary advantages for further improvement, which is our second contribution. Our third contribution is the scalability study of GMM and DNN-based approaches with respect to the number of speakers. The last contribution is our demonstration that the almost perfect state-of-the-art performance on the SSC corpus results from the specific
corpus setup. We show that the performance drops substantially on a standard speaker recognition corpus.

6.2 Insights

During the course of this dissertation study, we have gained a number of insights on robust SID. First and foremost, CASA proves to be an effective way of improving robustness of SID systems. A main computational goal of CASA is the IBM, which identifies target dominant T-F units and interference dominant ones. The incorporation of CASA to the GMM-based SID systems can be achieved in two ways. One is from the perspective of missing feature methods. As shown in this dissertation, bounded marginalization and feature reconstruction both benefit from well estimated CASA masks. The other perspective is to directly attenuate the interference in the T-F domain using CASA masks, and then either resynthesize to a time domain signal or extract cepstral features. Direct masking is such an example. On the other hand, it remains an interesting research problem on how to combine CASA masks and the DNN framework (Narayanan, 2014). Chapter 5 briefly touches upon this issue by creating training labels using the IBM. We discuss this problem in the next section.

Often robust SID methods employ different features and address the problem from different perspectives. Each of them has its own merits and limitations. A combination can thus be utilized to leverage potentially complementary advantages. A combination can be performed at the feature level, module level, system level, or together. As
illustrated in this dissertation, the combination idea consistently brings performance improvement over individual ones in various robust SID tasks.

The combined effects of additive noise and reverberation degrade speech intelligibility to a greater extent than either distortion for human listeners. We find similar trends in automatic speaker recognition. Our proposed method separately deals with the two distortions. Reverberant speaker models compensate for the mismatch problem created by reverberation and exhibit reasonable generalization to unseen $T_{60}$’s. Models of multiple training conditions are complementary, and thus a combination of their respective decisions achieves good performance to unseen test conditions. CASA-based speech separation continues to perform well against noise even in the presence of reverberation. Although IBM$_{DS}$ achieves the best SID performance, IBM$_R$ is most resistant to estimation errors.

DNN plays an important role in this dissertation. In Chapter 4, it is used for mask estimation, and generates quality masks that generalize well from simulated reverberation to real reverberation. We also employ DNN to directly perform cochannel SID in Chapter 5. Its performance is substantially better than the state-of-the-art GMM-based methods. Furthermore, our DNN-based approach demonstrates superior scalability to speaker set size compared to the GMM-based approach.
6.3 Future Study

A number of topics merit further investigation. Chapter 3 employs MLP classifiers for mask estimation. Recent CASA studies suggest that DNN classifiers produce better masks. Thus, it is worth studying whether DNN-estimated masks can further improve noise-robust SID performance. Meanwhile, the uncertainty decoding method used by Shao and Wang (2008) in feature reconstruction does not provide significant performance improvement due to errors of uncertainty estimation. Ideal uncertainty information, on the other hand, helps robust SID considerably. More accurate estimation of uncertainty information has the potential of boosting SID performance.

In Chapter 4, our study has found that speaker models trained using simulated RIRs do not generalize well to real RIRs. One possible explanation is that the derived frame level features are different for the two types of RIRs. Future studies should investigate how to bridge the gap and develop speaker models for both scenarios. Meanwhile, the DNN-based mask estimation for three reverberant IBM definitions exploits the combined feature set proposed by Wang et al. (2013). This feature set is proposed for anechoic speech separation. Further study is necessary to improve reverberant mask estimation. For instance, one can search for reverberation-robust features to feed into DNN classifiers.

In our cochannel SID study, we propose a DNN-based system. Since it is the first attempt of applying DNN for cochannel SID, we expect plenty of room for future improvement. For example, the training features and labels can be selected in a systematic way. Also, the architecture can be optimized to achieve a balance between
performance and computational complexity. Furthermore, both GMM and DNN-based approaches do not scale well to large speaker sets in reverberant conditions. Currently reverberation is accounted for by training in reverberant conditions. Additional preprocessing, such as speech dereverberation (Han et al., 2014), may be used to improve scalability in reverberant conditions.

Lastly, it is an exciting new direction to attempt to replace GMM-based SID framework by a DNN-based framework. One important question lies in how to incorporate time-frequency masking in a DNN-based approach. Our research in the cochannel case creates training labels using the IBM. Alternative ideas may include extracting IBM-based training features.
BIBLIOGRAPHY


