SUPERVISED SPEECH SEPARATION AND PROCESSING

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of the Ohio State University

By

Kun Han, M.S.

Graduate Program in Computer Science and Engineering

The Ohio State University

2014

Dissertation Committee:

Professor DeLiang Wang, Advisor

Professor Mikhail Belkin

Professor Eric Fosler-Lussier

© Copyright by

Kun Han

2014

ABSTRACT

In real-world environments, speech often occurs simultaneously with acoustic interference, such as background noise or reverberation. The interference usually leads to adverse effects on speech perception, and results in performance degradation in many speech applications, including automatic speech recognition and speaker identification. Monaural speech separation and processing aim to separate or analyze speech from interference based on only one recording. Although significant progress has been made on this problem, it is a widely regarded challenge.

Unlike traditional signal processing, this dissertation addresses the speech separation and processing problems using machine learning techniques. We first propose a classification approach to estimate the ideal binary mask (IBM) which is considered as a main goal of sound separation in computational auditory scene analysis (CASA). We employ support vector machines (SVMs) to classify time-frequency (T-F) units as either target-dominant or interference-dominant. A rethresholding method is incorporated to improve classification results and maximize hit minus false alarm rates.

Systematic evaluations show that the proposed approach produces accurate estimated IBMs.

In a supervised learning framework, the issue of generalization to conditions different from those in training is very important. We then present methods that require only a small training corpus and can generalize to unseen conditions. The system utilizes SVMs to learn classification cues and then employs a rethresholding technique to estimate the IBM. A distribution fitting method is introduced to generalize to unseen signal-to-noise ratio conditions and voice activity detection based adaptation is used to generalize to unseen noise conditions. In addition, we propose to use a novel metric learning method to learn invariant speech features in the kernel space. The learned features encode speech-related information and can generalize to unseen noise conditions. Experiments show that the proposed approaches produce high quality IBM estimates under unseen conditions.

Besides background noise, room reverberation is another major source of signal degradation in real environments. Reverberation when combined with background noise is particularly disruptive for speech perception and many applications. We perform dereverberation and denoising using supervised learning. A deep neural network (DNN) is trained to directly learn a spectral mapping from the spectrogram of corrupted speech to that of clean speech. The spectral mapping approach substantially attenuates the distortion caused by reverberation and background noise, leading to improvement of predicted speech intelligibility and quality scores, as well as speech recognition rates.

Pitch is one of the most important characteristics of speech signals. Although pitch tracking has been studied for decades, it is still challenging to estimate pitch from speech in the presence of strong noise. We estimate pitch using supervised learning, where probabilistic pitch states are directly learned from noisy speech data. We investigate two alternative neural networks modeling pitch state distribution given observations, i.e., a feedforward DNN and a recurrent deep neural network (RNN). Both DNNs and RNNs produce accurate probabilistic outputs of pitch states, which are then connected into pitch contours by Viterbi decoding. Experiments show that the proposed algorithms are robust to different noise conditions.

Dedicated to my mother, Minghui Zhang, and my father, Jiayou Han

ACKNOWLEDGMENTS

First and foremost, I would like to express the deepest appreciation to my advisor Dr. DeLiang Wang. He leads by example and generously teaches me the key qualities to become a researcher throughout my six-year Ph.D. study. I would like to thank him for his insights and guidance not only for my progress in the research projects, but also for the development of my personal career. All these skills are invaluable and will benefit the rest of my life.

Dr. Eric Fosler-Lussier has enlightened me on artificial intelligence in his class and broadened my knowledge of speech recognition. Dr. Mikhail Belkin has helped me better understand machine learning, which proves to be very useful in my research. I would also like to thank Dr. Eric Fosler-Lussier and Dr. Mikhail Belkin for serving on my dissertation committee and for providing valuable feedback on this dissertation. Dr. Brian Kulis has improved my understanding of graphical models and his work on metric learning has inspired my research. My thanks also go to Dr. Nicoleta Roman for her insightful suggestions on speech dereverberation. I appreciate the support and help from Dr. Tao Zhang, Dr. Bill Woods, and Dr. Ivo Merks from Starkey. Discussion with them on dereverberation is very insightful and benefits my research.

I am grateful to Dr. Ivan Tashev and Dr. Dong Yu of Microsoft Research, who supervised my internship and offered me the opportunity for a wonderful summer.

There are far too many to mention individually who have helped me in so many ways. To name a few, I thank Dr. Zhaozhang Jin for his patience in helping me to quickly start my research in the first few years. I thank Dr. Ke Hu for his suggestion on my research area and Dr. John Woodruff for answering my endless questions during my research. I have also benefited a lot from Arun Narayanan's skills on automatic speech recognition, and Xiaojia Zhao's expertise on speaker recognition; writing my dissertation in their company is helpful and inspiring. I also want to thank Yuxuan Wang for his knowledge on deep learning and I have the pleasure to co-author a few papers with him, which are described in this dissertation. Many thanks are due to Donald Williamson and Jitong Chen for our interesting discussion on our research. I also thank Dr. Chao-Ling Hsu, Dr. Zhiyao Duan, Dr. Jian Tang, Dr. Yi Jiang, and Dr. Xiao-Lei Zhang for helping me improve my understanding in different areas.

I am grateful to my friends I met when I first came to the United States. They make my life joyful and entertaining.

My most heartfelt acknowledgment must go to my family. They have always given me the utmost support in any endeavor. Without their persistent support, the journey in my life would not have been possible.

I wish to acknowledge financial support me from two AFOSR grants (FA9550-08-1-0155 and FA9550-12-1-0130), an NIDCD grant (R01 DC012048), and Starkey.

VITA

Nov, 1982	Born - Chuzhou, China
2004	B.E. in Electronic Engineering, Nanjing University of Aeronautics and Astronautics
2008	M.S. in Computer Science, University of Science and Technology of China

PUBLICATIONS

K. Han and D. Wang. Neural networks for supervised pitch tracking in noise. In *Proc. IEEE ICASSP*, pages 1502–1506, 2014.

K. Han, Y. Wang, and D. Wang. Learning spectral mapping for speech dereverberation. In *Proc. IEEE ICASSP*, pages 4661–4665, 2014.

K. Han and D. Wang. Learning invariant features for speech separation. In *Proc. of IEEE ICASSP*, pages 7492–7496, 2013.

K. Han and D. L. Wang. Towards generalizing classification based speech separation. *IEEE Trans. Audio, Speech, and Lang. Process.*, 21(1):166–175, 2013.

K. Han and D. L. Wang. On generalization of classification based speech separation. In *Proc. of IEEE ICASSP*, pages 4541–4544, 2012.

Y. Wang, K. Han, and D. L. Wang. Exploring monaural features for classification-based speech segregation. *IEEE Trans. Audio, Speech, Lang. Process.*, 21(2):270–279, 2012.

Y. Wang, K. Han, and D. L. Wang. Acoustic Features for Classification Based Speech Separation. In *Proc. Interspeech*, pages 1532–1535, 2014.

K. Han and D. L. Wang. A classification based approach to speech segregation. *J. Acoust. Soc. Am.*, 132(5):3475–3483, 2012.

K. Han and D. L. Wang. An SVM based classification approach to speech separation. In *Proc. of IEEE ICASSP*, pages 5212–5215, 2011.

FIELDS OF STUDY

Major Field: Computer Science and Engineering

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

1.1 Motivation

When I was writing this dissertation, a science fiction action film *Transformers 4: Age Of Extinction* was released. The plot of the film is rather lame and disappointing. What I can remember are 165 minutes of explosions, car crashes, cars turning into robots, and human figures shouting. I sat in the cinema and complained about the film to my friends. Although we talked in a low voice and the background sounds were strong, we could heard each other very well. How wonderful our human auditory system is!

This is a very common situation we face in our daily life. The speech sound reaches our ears is usually not just a clean utterance, but a mixture with acoustic interference, such as movie sound, music, traffic noise, or another speech utterance. In such a situation, a normal-hearing listener excels at separating the target sound from various types of interference. Cherry used the term “cocktail party effect” to describe the phenomenon of being able to focus auditory attention on a single conversation while filtering out a range of other sound sources in a noisy room [17]. However, comparing to the auditory system is a remarkable capacity, speech separation is still a great challenge for machines.

Monaural speech separation and processing are the tasks of separating or analyzing a speech signal from a monaural recording when the background interference presents. For this task, the information regarding sound directions is not available, and one can only make use of the intrinsic acoustic properties of speech and interference. The task has proven to be extremely challenging [134]. On the other hand, dealing with interference is strongly needed in many speech applications, for example, automatic speech recognition (ASR) [81], and speaker identification (SID) [31], [149].

Various approaches have been proposed for monaural speech separation and processing, including speech enhancement [45], [62] and model based approaches [111], [6], [103]. However, these methods either need strong assumptions regarding the statistical properties of interference or rely on pretrained source models.

Psychoacoustic research in *auditory scene analysis* (ASA) [14] suggests that auditory segregation has two stages: segmentation and grouping. In segmentation, the input sound is decomposed into sensory elements (or segments), each of which should originate from a single source. In grouping, the segments that likely arise from the same source are aggregated together. The cues that characterize intrinsic sound properties, including harmonicity, onset and offset, play a pivotal role in segregation. Inspired by the principles of ASA, *computational auditory scene analysis* (CASA) aims to utilize auditory cues to segregate target sound from interference [134].

Machine learning, as a branch of artificial intelligence, focuses on the construction and study of systems that can learn from data. In recent years, the machine learning and speech processing communities have had increasing influences on each other. From a machine learning standpoint, the speech separation problem is to infer clean speech from noisy speech signals. We believe it is reasonable to formulate the separation problem as a supervised learning problem, i.e., given features extracted from noisy speech, we train a model to predict clean speech outputs. A typical example is the concept of the ideal binary mask (IBM), which has been suggested as a main computational goal for CASA systems [133]. With the target of the IBM, the speech separation problem is formulated as a classification problem. In addition, it is natural to train a model to directly learn a spectral representation of clean speech from its noisy version, a case of training a regression model. We will develop these approaches in this dissertation.

Further, deep learning has emerged as a new trend in machine learning since 2006 [50], [49]. A deep neural network (DNN) is a feedforward neural network that has more than one hidden layer between its input and output. It is capable of utilizing large-scale data and learning high-level representations from raw features. These advances enable effective modeling of nonlinear interactions between speech and the acoustic environments as well as dynamic structure of speech. With sufficient data and appropriate training strategies, DNNs perform very well in many machine learning tasks, such as, speech recognition [88], [48], and speech separation [140].

Motivated by recent progress, we will employ deep learning to address the speech separation and processing problems in this dissertation.

1.2 Computational objectives

This dissertation is concerned with monaural speech separation and processing from nonspeech interference. The goal of this dissertation is to build a robust speech separation and processing system.

As we have mentioned above, a main goal for CASA systems is the IBM. The IBM is defined in terms of premixed target and interference. Specifically, with a time-frequency (T-F) representation of a sound mixture, the IBM is a binary matrix along time and frequency where a matrix element is 1 if the signal-to-noise ratio (SNR) within the corresponding T-F unit is greater than a local SNR criterion (LC) and is 0 otherwise. A series of recent studies shows that IBM separation produces large speech intelligibility improvements in noise for both normal-hearing and hearing-impaired listeners [3], [15], [80], [136].

Adopting the IBM as the computational goal, we can formulate sound separation as binary classification. A support vector machine (SVM) aims to find an optimal (i.e., largest margin) hyperplane to classify data [127], which is successfully applied to many classification problems. Typically, the output of the discriminant function of an SVM is a real number, the absolute value of which indicates the distance from the optimal hyperplane. The threshold of 0 is usually used to binarize the output to calculate the label of each datum. However, the default threshold is not always an optimal choice when the measurement is not classification accuracy. In speech separation, it has been shown that the difference between the hit rate (HIT) and the false alarm rate (FA) is well correlated to speech intelligibility [70]. Therefore we introduce rethresholding to adjust thresholds to achieve higher HIT−FA rates rather than classification accuracies.

For supervised learning to be effective, the distribution of the training set needs to match that of the test set. For speech separation, if input SNRs or background noises in test mixtures are not seen in the training set, the trained classifier will unlikely achieve good classification results. Previous systems have avoided this issue by testing on SNR and noise conditions similar to those in training. Hence, it is important to investigate the generalization capability of such classifiers. We observe that, with SVM based classification, rethresholding can significantly improve estimated IBMs under new noisy conditions. Therefore, we convert generalization to a threshold determination problem. The new thresholds are adaptively computed based on the characteristics of test mixtures, and they are expected to generalize to new SNR or noise conditions. The rethresholding based approach yields good generalization performance, but it is difficult to apply it to real-time applications because rethresholding is applied after a system sees a certain portion of test signals. A more desirable approach directly focuses on the mismatch problem of the training set and the test set. We use a feature transformation approach to project original features to a new space such that new features are robust to different noises. A model trained on these new features is able to generalize to new noisy conditions.

Room reverberation introduces distortion in both temporal and spectral domain. Although human listeners have an extraordinary ability of separating a sound of interest from its background under reverberant conditions, it is very challenging for machines to simulate this perceptual ability. In this scenario, we aim to separate clean anechoic speech from a reverberant noisy mixture. We use supervised learning to acquire a spectral mapping for dereverberation and denoising. With strong learning capacity, a DNN is expected to be able learn the mapping in the magnitude spectrogram domain. Therefore, we convert dereverberation and denoising to a regression problem, i.e., given a real-valued noisy magnitude spectrogram, we train a model to estimate a real-valued clean magnitude spectrogram.

One important characteristic of speech is the fundamental frequency ($F0$), or pitch. It has been shown that pitch information benefits many speech applications, including speech separation, recognition, speaker identification. Extracting pitch from noise is difficult, because the harmonic structure is corrupted by noise. To determine pitch in noise, we use supervised learning to estimate pitch frequency in each frame. Utilizing temporal dependency of pitch, we connect frame-level pitch points into pitch contours using sequential modeling.

1.3 Background

We now introduce basics of speech enhancement and model-based techniques for speech separation and survey related studies.

1.3.1 Spectral subtraction

Speech enhancement is concerned with improving perceptual aspects of speech that has been degraded by additive noise. In most applications, the objective of speech enhancement is to improve the quality and intelligibility of degraded speech [83]. In this case, speech enhancement algorithms mainly deal with additive noise, instead of competing speech. Therefore, these approaches can utilize the characteristics of speech to reduce or suppress the background noise.

The spectral subtraction method is probably the earliest speech separation method in real-world applications. The principle of spectral subtraction is simple: the clean signal spectrum can be estimated by subtracting the estimate of the noise spectrum from the noisy speech spectrum. This method usually uses nonspeech intervals to estimate the noise spectrum. Therefore, it requires the assumption that the noise spectrum does not change significantly in the time domain. Consider that clean speech $x(n)$ is corrupted by additive noise $d(n)$. The mixture $y(n)$ is,

$$y(n) = x(n) + d(n) \tag{1.1}$$

Then we take the discrete-time Fourier transform for both sides:

$$Y(\omega) = X(\omega) + D(\omega) \tag{1.2}$$

where, $Y(\omega)$ can be expressed in terms of its magnitude and phase as $Y(\omega) =$

$|Y(\omega)|e^{j\phi_y(\omega)}$. As phase is not expected to affect speech intelligibility, we can replace the speech phase and noise phase with the noisy speech phase. Therefore, we have

$$|X(\omega)| = |Y(\omega)| - |D(\omega)| \tag{1.3}$$

In practice, $|Y(\omega)|$ can be calculated from the mixture. Then one needs to estimate the noise magnitude spectrum $|D(\omega)|$, which is usually replaced by its average value computed during nonspeech intervals. Note that, due to inaccuracies introduced by the noise spectrum estimation, $|X(\omega)|$ could be negative according to Eq. 1.3. In this case, one simple solution is to set $|X(\omega)|$ to 0 to ensure a nonnegative magnitude spectrum.

It is easy to implement a spectral subtraction method to reduce the noise in the corrupted signal. However, a few drawbacks exist. As we discussed above, if the left side of Eq. 1.3 is negative, we can directly set it to 0, which introduces some isolated peaks in the spectrum domain. In the time domain, these peaks sound similar to tones and are commonly referred to as *musical noise*.

Some studies utilize the oversubtraction technique to overcome the shortcoming of speech distortion [9], [12]. That is, when subtracting the estimate of the noise spectrum from the noisy speech spectrum, we oversubtract the noise spectrum to further reduce the amplitude of peaks and use nonzero spectral floor to fill in the spectral valleys. Berouti *et al.* [9] found that noisy speech processed by oversubtraction had less musical noise than that processed by original subtraction. Further, studies use more flexible methods to perform oversubtraction, including nonlinear spectral [82] and multiband spectral subtraction [68], where different frequencies or different subbands have different subtraction factors.

Another problem for spectral subtraction is that it requires noise to be stationary or slowly varying. For nonstationary noise, spectral subtraction is not able to effectively estimate noise spectrum from nonspeech intervals and thus the approach does not work well.

1.3.2 Wiener filtering

Spectral subtraction estimates speech spectra using the instantaneous spectra of the noisy signal and the time-averaged spectra of the noise. If the power spectra of speech can be estimated, one can design a filter to restore speech.

The Wiener filter derives the enhanced speech by minimizing the mean-square error in *complex spectrum* domain. Let $H(\omega)$ be the discrete Fourier transform (DFT) of the impulse response of the filter. Then the spectrum of speech can be computed by

$$X(\omega) = H(\omega) \cdot Y(\omega) \tag{1.4}$$

To obtain $H(\omega)$, we have:

$$H(\omega) = \frac{P_x(\omega_k)}{P_x(\omega_k) + P_d(\omega_k)} = \frac{\xi_k}{\xi_k + 1} \tag{1.5}$$

where, $P_x(\omega)$ and $P_d(\omega)$ are the power spectrum of $x(n)$ and $d(n)$, respectively. ξ_k is called *a priori* SNR, defined as the ratio of $P_x(\omega)$ and $P_d(\omega)$, which can be estimated as a weighted combination of the past and present estimates of ξ_k [27], [116].

Compared with spectral subtraction, the noise residual of noisy speech processed by Wiener filtering is similar to white noise, which is more tolerable than music noise for human listeners. The original Wiener filter assumes stationary noise, and this assumption can be relaxed to nonstationary noise by using Kalman filters [98]. Wiener filters are considered to be linear estimators of the clean signal spectrum, because in the frequency domain the enhanced speech spectrum is obtained by multiplying the noisy speech spectrum by a Wiener filter. Some nonlinear estimators, such as statistical-based methods, could potentially yield better performance.

1.3.3 MMSE estimation

Spectral subtraction is an estimator with little or no assumptions about the prior distributions for power spectra of speech and noise. In fact, speech and noise signal usually have their statistic characteristics and one can utilize prior statistical distributions to design an estimator.

Minimum mean square error (MMSE) estimators have been developed under various assumptions such as Gaussian sample distributions, lognormal distribution of spectral magnitudes, etc. MMSE approach aims to minimize the mean-square error (MMSE) between the estimated and true *spectral magnitudes*:

$$e = E[(\hat{X}_k - X_k)^2] \tag{1.6}$$

where $\hat{X}_k$ is the estimated spectral magnitude at frequency ω_k, and X_k is the true magnitude of the clean signal. Given observed noisy speech $\mathbf{Y} = [Y(\omega_0)Y(\omega_1)\ldots Y(\omega_{N-1})]$, the optimal MMSE estimator is:

$$\hat{X}_k = E[X_k|\mathbf{Y}] = \int X_k p(X_k|\mathbf{Y})dX_k \tag{1.7}$$

If we have prior knowledge about the distributions of the speech and noise DFT coefficients, we can evaluate the mean of the posterior probability density function of X_k, i.e., $p(X_k|\mathbf{Y})$.

Note that, similar to Wiener filtering, the MMSE estimator assumes that the *a priori* SNR and the noise power spectrum are known. In practice, however, their estimation is not trivial, because one can only access the noisy speech. Ephraim and Malah [27] assumed that the Fourier transform coefficients have a complex Gaussian probability distribution. Based on this assumption, one can calculate the conditional probability density function $p(Y(\omega_k)|X_k)$ and the prior $P(X_k)$. By using Bayesian formula, one can obtain the estimate of X_k. They proposed a method to estimate the *a priori* SNR [27], where the speech power spectrum is computed by a maximum-likelihood method and the noise power spectrum is estimated during nonspeech intervals. However, if noise stationarity does not hold, it is not easy to obtain the noise power spectrum. Hendriks *et al.* [45] proposed a noise power spectrum estimation algorithm, which uses a weighting function derived from MMSE to estimate the noise power spectrum, which performs well for both stationary and nonstationary noises. In addition, Jensen and Hendriks [62] derived a gain function based on the same spectral magnitude MMSE as in Hendriks *et al.* but generated an optimal binary mask in the MMSE sense, which is a binarization based on gain thresholds. Their MMSE based continuous masks can improve speech intelligibility to some extent.

A key assumption made in the above MMSE algorithms is that the real and imaginary parts of the clean DFT coefficients follow a complex Gaussian distribution. Such an assumption, however, may not hold in some situations. It has been shown that Gamma and Laplacian probability distributions provide a better fit to the experimental data than the Gaussian distribution [83].

Statistical model based speech separation analyzes the statistical properties of speech and noise signal. Therefore, in general it is not sensitive to speakers. As can be seen, it is important to estimate critical quantities such as the *a priori* SNR or the noise power. However, the estimation of these parameters depends on statistical models, and it is questionable whether the statistical assumptions are applicable to different noisy conditions.

1.3.4 Model based methods

The previous approaches can be categorized as speech enhancement approaches, which aim to either enhance speech or attenuate noise in noisy speech. If the interference is competing speech, speech enhancement algorithms are not able to separate them. Model based methods use generative models to capture the feature statistics of isolated sources, and thus source separation becomes the problem of using prior models to identify a set of source signals that combine to produce the observed mixture signal [25].

Due to the temporal continuity of speech, it is naturally to use hidden Markov models (HMMs) to model speech signals. Roweis's system [111] trained an HMM using narrowband spectrogram for each speaker. To separate a mixture of multiple known speakers, these pretrained models are combined into a factorial HMM architecture and separation is done by inferring an underlying state sequence of the multiple Markov chains.

A large number of studies formulated speech separation as a non-negative matrix factorization (NMF) problem, where the spectrogram or cochleagram of a signal $\mathbf{Y}$ can be represented as

$$\mathbf{Y} = \mathbf{B} \cdot \mathbf{G} \tag{1.8}$$

where, $\mathbf{B}$ is the basis matrix and $\mathbf{G}$ is the encoding matrix. For a multiple-source mixture, $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_K]$ and $G = [\mathbf{G}_1, \ldots, \mathbf{G}_K]^T$, where $\mathbf{B}_k$ and $\mathbf{G}_k$ correspond to the basis matrix and the encoding matrix of the kth source. If one can decompose $\mathbf{Y}$ to the multiplication of $\mathbf{B}$ and $\mathbf{G}$, it is straightforward to reconstruct the kth source by $\mathbf{X}_k = \mathbf{B}_k\mathbf{G}_k$.

Lee and Seung [78] proposed to decompose the matrix by minimizing the reconstruction error between the observation $\mathbf{Y}$ and the model $\mathbf{BG}$. Virtanen [132] incorporated temporal continuity to the cost function and iteratively updated the gains and the spectra.

The model-based approaches rely on pretrained models, resulting in a difficulty on generalization. Researchers attempted to overcome this problem by model adaptation. Ozerov *et al.* [97] proposed a general framework for model based source separation, which can be applied to either blind separation with random initialization or non-blind separation with pretrained models. But without prior knowledge, random initialization usually does not yield satisfactory performance according to on our experiments.

1.4 Organization of dissertation

The rest of this dissertation is organized as follows.

The next chapter presents a classification based speech separation to estimate the IBM. This study employs support vector machines to classify T-F units as either target-dominant or interference-dominant. A rethresholding method is incorporated to improve classification results and maximize hit minus false alarm rates. An auditory segmentation stage is utilized to further improve estimated masks. Systematic evaluations show that the proposed approach produces high quality estimated IBMs and outperforms another classification based separation system in terms of classification accuracy.

Chapter 3 investigates the generalization problem for classification based speech separation. This study focuses on the situation of mismatch between the training set and the test set. This chapter presents methods that require only a small training corpus and can generalize to unseen conditions. The system utilizes SVMs to learn classification cues and then employs a rethresholding technique to estimate the IBM. A distribution fitting method is used to generalize to unseen SNR conditions, and voice activity detection based adaptation is used to generalize to unseen noise conditions.

Chapter 4 describes a different approach to address the generalization problem. We propose to use a metric learning method to extract invariant speech features in the kernel space. As the learned features encode speech-related information that is robust to different noise types, the system is expected to generalize to unseen noise conditions.

Chapter 5 presents a DNN based approach for dereverberation and denoising. The input is a magnitude spectrogram of noisy speech and the output is that of clean speech. A DNN is trained to learn a spectral mapping to remove or attenuate reverberation and noise. We evaluate our approach for dereverberation, speech separation, and ASR tasks.

Chapter 6 discusses pitch estimation in noise. We investigate two alternative neural networks modeling pitch state distribution given observations. The first one is a feedforward DNN, which is trained on static frame-level acoustic features. The second one is a recurrent deep neural network (RNN) which is trained on sequential frame-level features and capable of learning temporal dynamics. Both DNNs and RNNs produce accurate probabilistic outputs of pitch states, which are then connected into pitch contours by Viterbi decoding.

Chapter 7 summarizes the contributions of this dissertation and outlines future research directions.

CHAPTER 2

A CLASSIFICATION BASED APPROACH TO SPEECH SEPARATION

For monaural speech separation, one can only utilize the intrinsic properties of speech or interference to separated target speech from background noise. The IBM has been proposed as a main goal of sound separation in CASA, and has led to substantial improvements of human speech intelligibility in noise. This chapter proposes a classification approach to estimate the IBM, and employs support vector machines to classify T-F units as either target-dominant or interference-dominant. A rethresholding method is incorporated to improve classification results and maximize hit minus false alarm rates. An auditory segmentation stage is utilized to further improve estimated masks. Systematic evaluations show that the proposed approach produces high quality estimated IBMs and outperforms a recent system in terms of classification accuracy. The work presented in this chapter has been published in the *Proceedings of the 2011 IEEE International Conference on Acoustic, Speech, and Signal Processing* [36] and *Journal of the Acoustical Society of America* [37].

2.1 Introduction

Monaural speech separation has been studies for decades. Various approaches have been proposed for monaural speech separation. Speech enhancement approaches [27], [45], [62] utilize the statistical properties of the signal to enhance speech that has been degraded by additive non-speech noise, which need assumptions regarding the statistical properties of signals. Model based approaches [115], [96], [148], [97] use trained models to capture the characteristics of individual signals for separation but they strongly rely on pre-trained source models. On the other hand, computational auditory scene analysis (CASA) [134] aims to separate a sound mixture based on perceptual principles [14].

As we mentioned in Sect. 1.2, the IBM has been suggested as a main goal for CASA systems [133], which is defined in terms of premixed target and interference and shown to produces substantial speech intelligibility improvements in noise for both normal-hearing and hearing-impaired listeners [3], [15], [80], [136].

Since the IBM is a matrix of binary values, IBM estimation is a form of binary classification. Roman *et al.* [107] proposed an early supervised classification method for IBM estimation although the method used binaural features for speech separation. Several studies employ binary classification for IBM estimation in the monaural domain. Seltzer *et al.* [118] treated the identification of noise components in a spectrogram as a Bayesian classification problem for robust automatic speech recognition. Weiss and Ellis [141] utilized relevant vector machines to classify T-F units. Jin and Wang [63] trained multilayer perceptrons (MLP) to classify T-F units using pitch-based features. Their system obtains good separation results in reverberant conditions. Kim *et al.* [70] used Gaussian mixture models (GMM) to learn the distribution of amplitude modulation spectrum (AMS) features for target-dominant and interference-dominant units and then classified T-F units by Bayesian classification. Their classifier led to speech intelligibility improvements for normal-hearing listeners. Kim and Loizou [69] further proposed an incremental training procedure to improve speech intelligibility, which starts from a small initial model and updates the model parameters as more data become available.

From the classification point of view, the first issue to address is feature extraction. The features used should distinguish target-dominant units from interference-dominant units. Pitch, or harmonic structure, is a prominent feature in voiced speech. Some previous studies show that pitch-based features are very effective for IBM estimation and robust to various forms of signal corruption [54], [63]. However, pitch-based features cannot address unvoiced speech separation because unvoiced speech lacks harmonic structure. On the other hand, AMS contains information for discriminating both voiced and unvoiced speech from nonspeech intrusions [126], [70]. We propose to combine these two types of features and construct a larger feature set for classification, which is expected to be discriminative in both voiced and unvoiced speech and generalize to different noise types.

Another important issue for classification is classifier design. Previously, MLPs [56], [63] and GMMs [70] have been explored for classification based speech separation.

In this study, we propose to use SVMs, which find an optimal (i.e., largest margin) hyperplane to classify data [127]. Typically, the output of the discriminant function of an SVM is a real number, the absolute value of which indicates the distance from the optimal hyperplane. The threshold of 0 is commonly used to binarize the output to calculate the label of each datum. In this study, we introduce a rethresholding technique to improve classification results and maximize a different measure called the hit rates minus false-alarm rates. In addition, we incorporate an auditory segmentation method to group more target-dominant units and remove interference-dominant units [63].

The chapter is organized as follows. In the next section, we present an overview of the proposed system. Section 2.3 describes how to extract auditory features. A detailed description of SVM classification is presented in Section 2.4. Section 2.5 describes the auditory segmentation stage. The systematic evaluation results and comparison are given in Section 2.6. We discuss related issues and conclude the chapter in Section 2.7.

2.2 System overview

Fig. 2.1 shows the diagram of the proposed system, which consists of several stages. The first stage of the system is auditory peripheral analysis. An input mixture signal $x(t)$ is resampled to 16000 Hz and analyzed by a 64-channel gammatone filterbank, with their center frequencies distributed from 50 Hz to 8000 Hz [135]. This filterbank is a standard model of cochlear filtering and is derived from psychophysical studies of

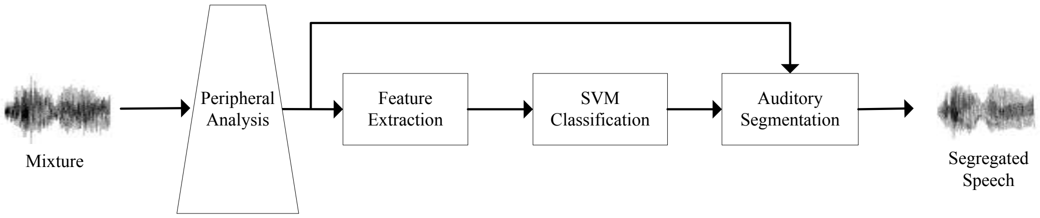


Figure 2.1: Diagram of the proposed speech separation system.

the auditory periphery [99]. In each channel, the output is divided into 20-ms time frames with 10-ms overlapping between consecutive frames. This processing produces a decomposition of the input signal into a two-dimensional T-F representation, or *cochleagram* [135]. Each T-F unit in the cochleagram corresponds to a frequency channel and a time frame.

The next stage, feature extraction, extracts two types of features from each T-F unit: pitch-based features [57] and AMS features [125]. After the feature extraction stage, we train SVMs to classify T-F units as either target-dominant or interference-dominant. Due to frequency specific characteristics of the input signal, one SVM is trained for each channel independently. Finally, in the auditory segmentation stage, we perform cross-channel correlation and onset/offset analysis to generate T-F segments. The T-F units in a segment primarily originate from the same sound source and therefore we group them into either the target or interference stream based on unit classification results. The final binary mask represents an estimate of the IBM and is used to resynthesize separated target speech. The resynthesis is basically performed by summing the filter responses in target-dominant units and compensating for phase shifts across the filterbank [135].

2.3 Feature extraction

2.3.1 Pitch-based features

Let $u_{c,m}$ denote a T-F unit for channel c and frame m and $x(c,t)$ denote the filter response for channel c at time t. To extract pitch-based features for $u_{c,m}$, the normalized autocorrelation function (ACF), $A(c,m,\tau)$, is computed at each lag τ [135]:

$$A(c,m,\tau) = \frac{\sum_n x(c,mT_m - nT_n)x(c,mT_m - nT_n - \tau T_n)}{\sqrt{\sum_n x^2(c,mT_m - nT_n)\sum_n x^2(c,mT_m - nT_n - \tau T_n)}} \tag{2.1}$$

Here, n denotes discrete time, $T_m = 10$ ms is the frame shift and T_n is the sampling time. We use input mixtures sampled at 16 kHz in this study, which gives $T_n = 0.0625$ ms. The above summation is over 20 ms, the length of a time frame. We also compute envelope ACF, $A_E(c,m,\tau)$, similar to Eq. (2.1), which captures amplitude modulation information in high frequency channels.

For voiced speech, $u_{c,m}$ is considered target-dominant if the corresponding response or response envelope has a period close to that of the target speech, i.e., pitch period $\tau_S(m)$ [54]. In this case, $A(c,m,\tau)$ will have a peak close to $\tau_S(m)$. Therefore, we can use the ACF and the envelope ACF at the pitch lag, $A(c,m,\tau_S(m))$ and

$A_E(c, m, \tau_S(m))$, to construct pitch-based features. These two features have been demonstrated to be effective for discriminating voiced speech [57].

As commonly done in automatic speech recognition, we calculate delta features in order to encode feature variations. Specifically, for $m \geq 2$, time delta feature $\Delta A^M(c, m, \tau_S(m))$ is simply set to $A(c, m, \tau_S(m)) - A(c, m-1, \tau_S(m))$; and $\Delta A^M(c, 1, \tau_S(m))$ is set to $\Delta A^M(c, 2, \tau_S(m))$ for convenience. We compute frequency delta feature $\Delta A^C(c, m, \tau_S(m))$ in the same way. The pitch-based feature vector is then given by:

$$\mathbf{x}_{ACF}(c, m) = \begin{pmatrix} A(c, m, \tau_S(m)) \\ A_E(c, m, \tau_S(m)) \\ \Delta A^T(c, m, \tau_S(m)) \\ \Delta A_E^T(c, m, \tau_S(m)) \\ \Delta A^C(c, m, \tau_S(m)) \\ \Delta A_E^C(c, m, \tau_S(m)) \end{pmatrix} \tag{2.2}$$

When we extract the pitch-based features, the pitch period $\tau_S(m)$ needs to be specified. In order to remove the influence of pitch errors on the speech separation system, we use *Praat* [10] to extract the ground-truth pitch from the premixed speech in the training phase.

In the test phase, we extract pitch from mixtures by a pitch tracker. Specifically, we use the recently proposed tandem algorithm [57] which iteratively estimates pitch and computes a binary mask. To further improve pitch tracking results, we generate the initial pitch estimate for the tandem algorithm by utilizing the multipitch tracker of [64] which works well when more than one voiced sound are present. The tandem algorithm produces accurate pitch estimation results under most conditions, but for some mixtures, the generated pitch contours overlap in the time domain. So we need to further group pitch contours into the target track. We first remove those pitch contours shorter than 50 ms or out of the plausible pitch range for the specific speaker; the plausible ranges of the female and male speakers are set to [150, 400 Hz] and [80, 300 Hz], respectively. For two overlapping pitch contours, we retain the one closer to the average pitch frequency (250 Hz for the female speaker and 130 Hz for the male speaker). To exclude residual interference pitch contours, we first employ a simple energy-based method to detect voiced frames. Specifically, we label a frame as strongly voiced if the normalized log energy of the frame is greater than 0.6, voiced if the energy is between 0.4 and 0.6, and unvoiced otherwise. Then a pitch contour is selected if more than 15% frames of this contour are strongly voiced or 35% frames are either voiced or strongly voiced. This simple selection method eliminates most interference pitch contours and produces the final pitch estimation result.

Note that, since unvoiced frames lack harmonic structure, we simply put 0 as the values of the corresponding vector. In this way, the pitch-based features will not play a role in unvoiced frames, and classification in those frames will instead rely on AMS features.

2.3.2 AMS features

AMS features exist in both voiced and unvoiced speech, which contain information on both center frequencies and modulation frequencies within each analysis frame [126]. We use the same method of AMS extraction described in [70]. Specifically, we first extract the envelope from the filter response within each T-F unit. The envelopes are computed by full-wave rectification and then decimated by a factor of 4. The decimated envelope is then Hanning windowed with zero-padding, and a 256-point fast Fourier transform (FFT) is computed. The FFT computes the modulation spectrum in each T-F unit, with a frequency resolution of 15.6 Hz. Next, the FFT magnitudes are multiplied by 15 triangular-shaped windows spaced uniformly across the 15.6-400 Hz range and summed to produce 15 modulation spectrum amplitudes, which represent the AMS feature vector. We denote them by $M_1(c, m), ..., M_{15}(c, m)$.

Similarly, we calculate delta features ΔM^T and ΔM^C across frames and channels respectively, as in [70]. The AMS feature vector is given by:

$$\mathbf{x}_{AMS}(c,m) = \begin{pmatrix} M_1(c,m) \\ ... \\ M_{15}(c,m) \\ \Delta M_1^T(c,m) \\ ... \\ \Delta M_{15}^T(c,m) \\ \Delta M_1^C(c,m) \\ ... \\ \Delta M_{15}^C(c,m) \end{pmatrix} \tag{2.3}$$

The total dimensionality of the AMS feature vector $\mathbf{x}_{AMS}(c,m)$ is $3 \times 15 = 45$. Finally, the pitch-based feature vector and the AMS feature vector are combined into a 51-dimensional feature vector for each T-F unit. The combined features are used as the input to the classifier.

2.4 SVM classification

Given the extracted features, the task now is to classify T-F units to either target-dominant or interference-dominant. As mentioned earlier, one SVM is trained for each filter channel. By applying a kernel trick, an SVM maps a feature vector $\mathbf{x}_i$ into a higher dimensional feature space where a hyperplane is derived to maximize the margin of class separation. In this study, we choose the radial basis function kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$.

In the training phase, given a set of pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is a feature vector and y_i is the corresponding binary label, the SVM requires a solution to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{2.4}$$

where $\mathbf{w}$ is the weight vector of the hyperplane. ξ is a nonnegative variable measuring the deviation of a data point from the hyperplane. C controls the trade-off between complexity of the SVM and the number of nonseparable points. $\boldsymbol{\phi}$ is the vector of a set of nonlinear functions which transform the input space to a feature space of higher dimensionality. b is the bias. The parameters C and γ must be specified, and we choose them using 5-fold cross-validation in each channel separately. The SVM library LIBSVM [16] is used in our experiments.

Once the SVM training is completed, we use the trained models to classify T-F units. The discriminant function for classification is given as follow:

$$f(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b = \sum_{i \in \mathrm{SV}} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{2.5}$$

where SV denotes the set of support vector indices in training data. α_i is a Lagrange multiplier, which can be determined in the training phase. For a textbook treatment of SVM, the reader is referred to [43].

The output of the discriminant function is a real number and the binary label of each datum is typically given by the sign of this output. We find that this standard method tends to under-label target-dominant units for several reasons. First, with unbalanced training samples, the SVM hyperplane is often skewed to the minority, i.e., the class with fewer data [1, 143]. For typical IBM estimation, the input SNR is around 0 dB and the interference is broadband noise. In this situation, target-dominant units are much fewer than interference-dominant units because the speech energy is more concentrated in the cochleagram than that of noise. The unbalanced data likely cause the trained SVMs to misclassify some 1s to 0s. The second reason is that we use different pitch trackers to extract pitch-based features in the training and test phases, which makes the hyperplane obtained from the training data not exactly match that of the test data. More discussion on this point will be given in Section 2.7. Additionally, the standard SVM aims to minimize the classification error, but one of the goals of this study is to maximize HIT−FA.

For the above reasons, we propose to apply rethresholding as a *post-training* strategy, which is used in the decision phase without affecting the training phase. This technique has been successfully used in some other applications [13], [120]. Given a feature vector $\mathbf{x}$, the discriminant function gives an algebraic distance from $\mathbf{x}$ to the optimal hyperplane [43]:

$$r = \frac{f(\mathbf{x})}{\|\mathbf{w}\|} \tag{2.6}$$

Therefore, those data with small $|f(\mathbf{x})|$ are close to the trained hyperplane and thus easy to be misclassified if the hyperplane is skewed. We adopt a channel-specific threshold to label $f(\mathbf{x})$. Specifically, we select the threshold θ_c that maximizes the HIT$-$FA rate in channel c in a validation set with 10 sentences, and then use the new threshold to binarize $f(\mathbf{x})$:

$$y(\mathbf{x}) = \begin{cases} 1, & \text{if } f(\mathbf{x}) > \theta_c \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

Other approaches can be used to adapt the hyperplane. For example, one can use $f(\mathbf{x})$ to estimate the *posteriori* probability $P(y = 1|f(\mathbf{x}))$, and use $P(y = 1|f(\mathbf{x})) = 0.5$ as a criterion to classify data [100]. Another method is to find the threshold which makes the percentage of each class matches the percentage in the training data. We have tried both methods, but they do not perform better than the simple cross-validation method.

With SVM classification, our system generates an estimated IBM by combining the classification results in all the channels. As an example, Fig. 2.2 illustrates the separation results for a noisy speech signal. Fig. 2.2(a) shows the cochleagram of a female utterance, “A man in a blue sweater sat at the desk,” from the IEEE corpus [110]. Fig. 2.2(b) shows the cochleagram of a factory noise. The cochleagram of their mixture at 0 dB is shown in Fig. 2.2(c). By comparing the energy of each T-F unit in Figs. 2.2(a) and (b), we obtain the IBM shown in Fig. 2.2(d) where 1 is indicated by white and 0 by black, and LC is -5 dB. Fig. 2.2(e) shows the binary mask generated by the standard SVMs without rethresholding. The SVMs correctly classify most T-F units in both voiced and unvoiced speech intervals, but miss some

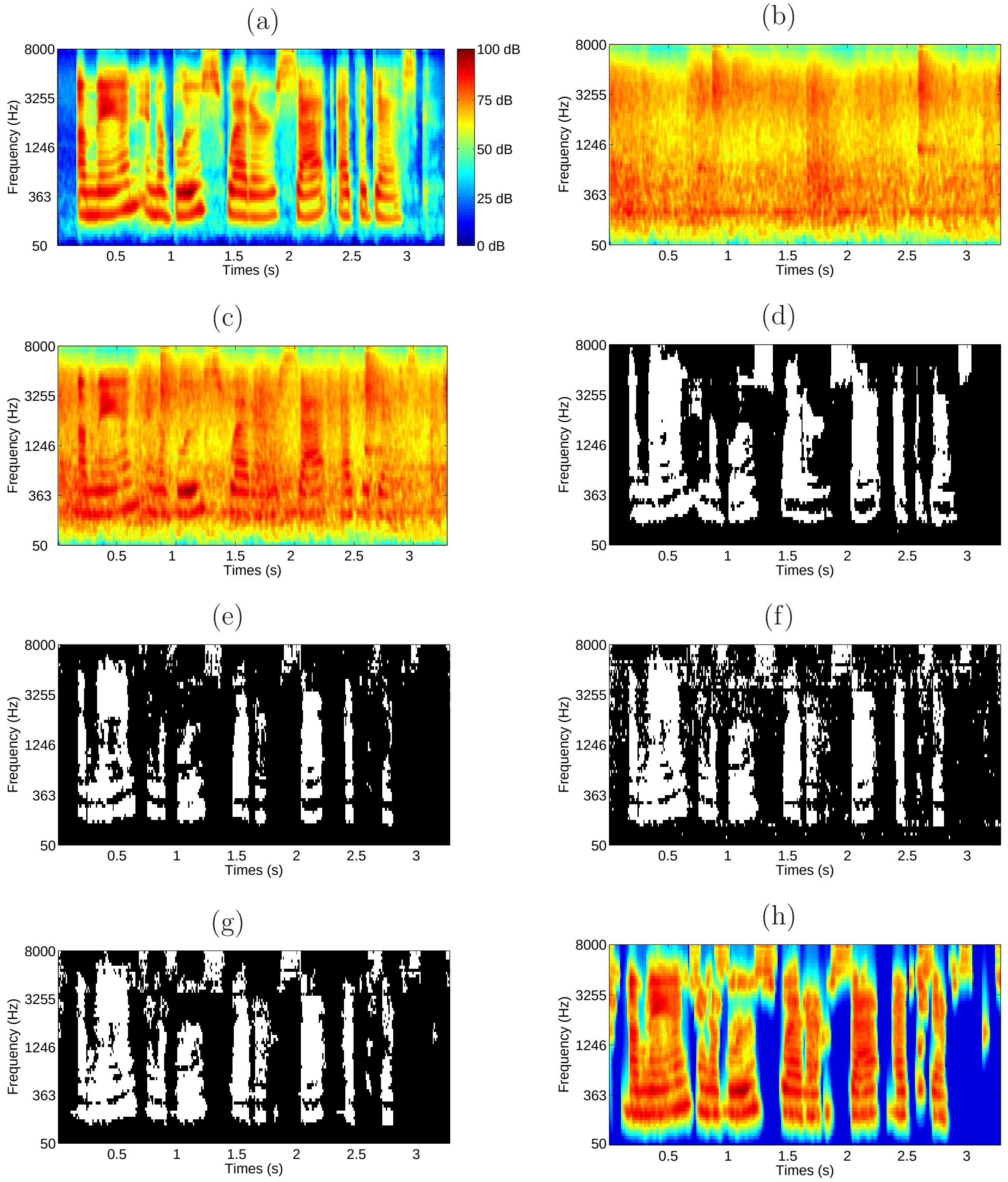


Figure 2.2: (Color online) IBM estimation. (a) Cochleagram of a female utterance. (b) Cochleagram of a factory noise. (c) Cochleagram of the mixture at 0 dB. (d) IBM for the mixture. (e) SVM-generated mask without rethresholding. (f) SVM-generated mask with rethresholding. (g) Estimated IBM after auditory segmentation. (h) Cochleagram of the masked mixture by the estimated IBM.

target-dominant units. By applying rethresholding, the system recovers many target-dominant units as shown in Fig. 2.2(f). This recovery comes at the expense of adding some scattered interference-dominant units.

2.5 Auditory segmentation

As shown in Fig. 2.2, an SVM-generated mask is close to the IBM, but still misses some target-dominant units and contains some interference-dominant units. We further improve estimated IBMs by auditory segmentation, which refers to a stage of processing that breaks the auditory scene into contiguous T-F regions each of which contains acoustic energy mainly from a single sound source [See also [63], [57], [58]].

With the voicing of a frame determined as described in Section III.A, we utilize cross-channel correlation to segment T-F units for voiced intervals [135]. The cross-channel correlation measures the similarity between the responses of two adjacent filters. The units with high cross-channel correlation indicate that they are likely from the same sound source. We calculate the cross-channel correlation of $u(c, m)$ as follow:

$$C(c, m) = \frac{1}{L} \sum_{\tau=0}^{L-1} \hat{A}(c, m, \tau) \hat{A}(c+1, m, \tau) \tag{2.8}$$

where $\hat{A}(c, m, \tau)$ denotes a normalized autocorrelation function with zero mean and unit variance, and L is the maximum delay for the plausible pitch frequency range from 70 to 400 Hz. For low frequency channels (below 2000 Hz), only units with sufficiently high cross-channel correlation (≥ 0.95) are iteratively merged into segments.

We use a similar way to calculate the cross-channel correlation of envelope response $C_E(c, m)$ and use it to segment units in high frequency channels (above 2000 Hz).

Since unvoiced speech lacks harmonic structure, we utilize onset/offset analysis [55] to segment T-F units within unvoiced intervals. Onsets and offsets correspond to sudden acoustic energy increases and decreases, respectively. Segments are formed by matching pairs of onset and offset fronts. In addition, a multiscale analysis is applied to integrate segments at several time-frequency scales [55].

With obtained segments, we first treat all the segments shorter than 50 ms (or 5 frames) as the interference. We then label each remaining segment wholly as the target (i.e. mask value 1) if more than half of the segment energy is included in the classified target units in Section 2.4. If a segment fails to be labeled as the target in this way, the individually classified T-F units within the segment are still included in the target stream. This results in the final estimated IBM, and the separated target speech can be resynthesized from this mask [135]. Fig. 2.2(g) shows a binary mask after auditory segmentation. We can see that most isolated interference-dominant units are removed from the mask and some missed target-dominant units are grouped at the same time. The cochleagram of the masked mixture by the estimated IBM is shown in Fig. 2.2(h). Note the similarity of Fig. 2.2(h) and Fig. 2.2(a).

2.6 Evaluation and comparison

2.6.1 Systematic evaluation

We evaluate the performance of our system by using the IEEE corpus [110], which contains 720 sentences spoken by two speakers, one male and one female. All utterances are downsampled from 25 kHz to 16 kHz. For the training set, we choose 100 utterances mixed with 3 types of noise—N1: speech-shaped noise, N2: factory noise, N3: 20-talker babble noise—at -5, 0 and 5 dB SNR. The test set consists of 60 utterances mixed with the 3 types of noise at -5 and 0 dB. There is no overlap between the training and the test utterances. Each utterance is mixed with a noise sample randomly cut out from the original noise recording. The LC is set to -5 dB for all 64 channels to generate IBMs. These choices are motivated by those in [70] where the same speech corpus and noises were used.

To quantify the performance of our system, we compute the HIT rate which is the percent of the target-dominant units in the IBM correctly classified, and the FA rate which is the percent of the interference-dominant units in the IBM wrongly classified. It has been shown that HIT$-$FA is highly correlated to human speech intelligibility [80], [70]. We also compute the classification accuracy, which is the percent of misclassified units.

Tables 2.1 and 2.2 show the average results for the female utterances and the male utterances, respectively. As shown in the tables, our system achieves relatively high

Table 2.1: Classification results for female utterances mixed with different noises at different input SNRs

		Speech-shaped		**Factory**		**Babble**	
		-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
Proposed	**HIT**	0.601	0.699	0.600	0.705	0.614	0.690
	FA	0.041	0.039	0.086	0.071	0.171	0.161
	HIT−FA	0.560	0.660	0.514	0.634	0.439	0.529
	Accuracy	0.903	0.896	0.861	0.870	0.775	0.786
Kim *et al.*	**HIT**	0.597	0.610	0.574	0.604	0.539	0.563
	FA	0.207	0.162	0.267	0.224	0.272	0.246
	HIT−FA	0.390	0.448	0.307	0.380	0.267	0.317
	Accuracy	0.763	0.782	0.706	0.731	0.684	0.689

Table 2.2: Classification results for male utterances mixed with different noises at different input SNRs

		Speech-shaped		**Factory**		**Babble**	
		-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
Proposed	**HIT**	0.541	0.674	0.561	0.667	0.590	0.660
	FA	0.098	0.081	0.153	0.125	0.234	0.192
	HIT−FA	0.442	0.593	0.408	0.542	0.355	0.468
	Accuracy	0.852	0.863	0.805	0.827	0.729	0.762
Kim *et al.*	**HIT**	0.573	0.576	0.545	0.558	0.460	0.491
	FA	0.195	0.150	0.241	0.204	0.251	0.225
	HIT−FA	0.379	0.427	0.304	0.354	0.210	0.266
	Accuracy	0.773	0.789	0.728	0.742	0.688	0.686

HIT rates and relatively low FA rates even at these low input SNRs. Under all conditions, the accuracy results are greater than 75% for the female utterances and 70% for the male utterances. These results demonstrate that our system produces high quality estimated IBMs. Here, the babble noise results are relatively lower than others, mainly because it is more difficult to group pitch contours under these conditions. We also observe that the pitch determination performance of the male utterances is slightly lower than that of the female utterances, causing the classification results for the male utterances not as good as those for the female utterances. We note that, without auditory segmentation, the average HIT−FA results in Tables I and II are lower by 2% for the female utterances and 5% for the male utterances.

In order to provide an indication of generalizability, we also test our system on two unseen noises, N4: white noise and N5: cocktail-party noise; different from the babble noise, the cocktail party noise mostly contains nonspeech background noise. Table 2.3 gives the results. From the table, one can see that our system achieves 58% HIT−FA rate for female speaker and 48% for male speaker on average, which are close to those with the noises in Tables 2.1 and 2.2. We believe that the generalizability of our system mainly results from the use of pitch-based features (See the following discussion associated with Table 2.6 and Section 2.6.2).

The proposed system utilizes pitch-based features and AMS features to classify T-F units. To investigate the relative merit of each feature type, we use each type to train a classifier. The training and the test corpora are the same as those for combined features. As pitch exists only in voiced speech intervals, the system with pitch-based

Table 2.3: Classification results for new noises

		Female Speaker				**Male Speaker**			
		White		**Cocktail-party**		**White**		**Cocktail-party**	
		-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
Proposed	**HIT**	0.694	0.726	0.543	0.663	0.718	0.771	0.460	0.616
	FA	0.073	0.083	0.070	0.063	0.167	0.177	0.146	0.152
	HIT-FA	0.622	0.642	0.473	0.600	0.551	0.595	0.314	0.464
	Accuracy	0.888	0.870	0.833	0.840	0.813	0.811	0.781	0.780
Kim *et al*	**HIT**	0.483	0.564	0.554	0.585	0.466	0.542	0.498	0.538
	FA	0.258	0.256	0.291	0.244	0.168	0.148	0.356	0.326
	HIT-FA	0.225	0.308	0.263	0.342	0.298	0.394	0.142	0.212
	Accuracy	0.698	0.700	0.670	0.696	0.768	0.777	0.617	0.634

features is trained only during voiced intervals. Similar to the system with combined features, the ground-truth pitch is used in the training phase and the estimated pitch is used in the test phase. For comparison, we evaluate HIT−FA results in voiced speech intervals which are determined by ground-truth pitch. Auditory segmentation is not included in all systems. Tables 2.4 and 2.5 compare the HIT−FA results for individual feature types. On average, the system with combined features achieves the best HIT−FA rate, which outperforms the AMS features by 3.3% and pitch-based features by 2.2%. Table 2.6 shows the comparison for new noises. In this case, the system with AMS features performs lower than that with combined features by around 20%. In contrast to AMS features, pitch-based features are robust to unseen noises, and achieve comparable results with combined features. This comparison suggests that the capacity of generalization of the proposed system mainly derives from pitch-based features. AMS features capture mixture envelopes which tend to be sensitive to different noises.

Table 2.4: Comparison of systems with different features for female utterances

HIT−FA	**Speech-shaped**		**Factory**		**Babble**	
	-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
Combined	0.518	0.632	0.504	0.603	0.436	0.462
AMS	0.506	0.600	0.433	0.527	0.412	0.462
Pitch-based	0.514	0.607	0.511	0.600	0.341	0.385

Table 2.5: Comparison of systems with different features for male utterances

HIT−FA	**Speech-shaped**		**Factory**		**Babble**	
	-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
Combined	0.369	0.535	0.376	0.511	0.346	0.430
AMS	0.388	0.446	0.373	0.426	0.363	0.412
Pitch-based	0.340	0.526	0.406	0.571	0.274	0.383

Table 2.6: Comparison of systems with different features for new noises

	Female Speaker				**Male Speaker**			
	White		**Cocktail-party**		**White**		**Cocktail-party**	
	-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
Combined	0.590	0.620	0.469	0.591	0.554	0.633	0.278	0.408
AMS	0.206	0.341	0.311	0.408	0.225	0.363	0.233	0.295
Pitch-based	0.603	0.642	0.382	0.552	0.608	0.674	0.258	0.397

Although our system is trained and tested on the IEEE corpus containing only one female and one male speaker, the classification system is expected to be speaker independent as the features used, i.e. AMS and pitch-based features, are not extracted in a speaker dependent way. To verify this, we directly use the trained models from the IEEE corpus, without change, to test on a new corpus from the TIMIT corpus [151] which contains different speakers. Specifically, for a system of each gender, the training set contains only one speaker from the IEEE corpus, but the test set contains 10 different speakers from the TIMIT corpus, each of which produces one utterance mixed with the three noises at -5 and 0 dB SNR. The test results on TIMIT utterances are given in Tables 2.7 and 2.8. As shown in the tables, although the test set uses different speakers, the separation results are only slightly lower than those shown in Tables 2.1 and 2.2. On average, there is 3.4% degradation for female speakers and 2.9% for male speakers in terms of HIT−FA rates, demonstrating that the system can generalize to different speakers. On the other hand, there is some gender dependency as male and female voices show distinct feature values (particularly pitch values). Gender dependency, however, is not a big limitation as one can readily train a male model and a female model, and gender detection is a relatively easy task [144].

2.6.2 Comparison with Kim *et al.*'s system

[70] proposed a speech separation system which obtains high HIT−FA rates for noisy IEEE utterances and demonstrates improved speech intelligibility in listening tests. Here, we compare our system with theirs in terms of HIT−FA. To implement their

Table 2.7: Classification results for female speakers on the TIMIT utterances

		Speech-shaped		**Factory**		**Babble**	
		-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
TIMIT	**HIT**	0.636	0.700	0.587	0.723	0.579	0.653
	FA	0.127	0.066	0.125	0.114	0.173	0.146
	HIT−FA	0.509	0.635	0.462	0.609	0.405	0.508
	Accuracy	0.835	0.875	0.824	0.845	0.773	0.788

Table 2.8: Classification results for male speakers on the TIMIT utterances

		Speech-shaped		**Factory**		**Babble**	
		-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
TIMIT	**HIT**	0.611	0.679	0.594	0.667	0.572	0.641
	FA	0.191	0.104	0.195	0.156	0.245	0.234
	HIT-FA	0.419	0.576	0.399	0.511	0.327	0.407
	Accuracy	0.777	0.841	0.768	0.798	0.713	0.723

system, we use AMS features to train a 256-component GMM for each binary label in each channel and test their system on the same corpus as used in evaluating our system. The results from their system are given in Tables 2.1-2.3.

From Tables 2.1 and 2.2, one can see that our system significantly outperforms theirs in terms of HIT−FA and accuracy. The average improvements are 17% for the HIT−FA rate and 9% for accuracy. Table 2.3 shows that their system does not generalize well to the two unseen noises, where the HIT−FA rates obtained are all lower than 40%. We have computed 95% confident intervals of HIT−FA means under all conditions, all of which are less than 2.5% for the proposed system and 2% for Kim *et al.*'s system. These analyses show that the performance differences are statistically significant.

As we have seen above, these comparisons show that our system significantly outperforms Kim *et al.*'s system. We should point out that the amount of training data used in the above comparison may be inadequate for the GMM classifiers used in Kim *et al.*, which have more parameters than the SVM classifiers used in our system. In addition, their system uses a 25-channel frontend and the above comparison uses a 64-channel frontend. While the reliance on a large amount of training data should be considered as a limitation, these differences nonetheless may put Kim *et al.*'s system in an unfavorable situation. To rectify this situation, we perform a further comparison using exactly the same frontend processor, same features, and same training methodology as in Kim *et al.*, except for the classifiers. Specifically, we first downsample utterances from 25 kHz to 12 kHz, and then use the 25-channel mel-scale filterbank as in Kim *et al.*'s system. Only AMS features are extracted from each T-F unit. The training set includes 390 IEEE sentences, each of which is mixed with the 3 noises at 3 input SNRs as described in the previous subsection. The test set includes 60 sentences mixed with 3 noises at -5 and 0 dB. The LC is set to -8 dB for the lower 15 frequency channels and -16 for the higher 10 frequency channels. No auditory segmentation is applied in our system. For a rigorous comparison, we train our SVM-based system and directly use the program code with trained GMMs provided by them to estimate the IBM.

Tables 2.9 and 2.10 show the comparative results. Our system obtains greater than 60% HIT$-$FA rates for the female utterances and greater than 50% HIT$-$FA rates for the male utterances. Compared to GMMs, SVMs improve HIT$-$FA rates under most

Table 2.9: Classification results with AMS features for female utterances

		Speech-shaped		Factory		Babble	
		-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
SVM	**HIT**	0.775	0.829	0.743	0.823	0.808	0.831
	FA	0.084	0.101	0.129	0.139	0.158	0.166
	HIT−FA	0.691	0.728	0.614	0.684	0.650	0.665
GMM	**HIT**	0.808	0.796	0.819	0.814	0.814	0.784
	FA	0.133	0.147	0.240	0.219	0.166	0.164
	HIT−FA	0.676	0.650	0.580	0.595	0.648	0.620

Table 2.10: Classification results with AMS features for male utterances

		Speech-shaped		Factory		Babble	
		-5 dB	**0 dB**	**-5 dB**	**0 dB**	**-5 dB**	**0 dB**
SVM	**HIT**	0.677	0.748	0.652	0.753	0.766	0.768
	FA	0.067	0.071	0.152	0.154	0.183	0.163
	HIT−FA	0.611	0.678	0.501	0.598	0.563	0.605
GMM	**HIT**	0.760	0.763	0.769	0.770	0.777	0.757
	FA	0.158	0.156	0.262	0.234	0.220	0.196
	HIT−FA	0.602	0.608	0.508	0.535	0.558	0.561

conditions, except for the factory noise at -5 dB for the male utterances where results are comparable. Statistically, the 95% confident intervals of the HIT−FA means for the proposed system are around ±1.5%, while those for the GMM system are ±2% on average.

2.7 Discussion and conclusion

In this study, we have proposed SVM-based classification for IBM estimation. As a discriminative classifier, the SVM does not model the distribution of the observed features but directly gives a predictive model conditioned on the observed data. The SVM aims to not only minimize the classification error but find a hyperplane with the largest margin, which potentially improves generalizability. In contrast, the GMM specifies a joint probability density function over observed data and labels, and tends to make more assumptions than discriminative classifiers. We also attempted to use MLPs as classifiers, but observed that the performance is poorer than either that of SVMs or GMMs.

By using rethresholding, we obtain improved classification results. As standard SVMs tend to under-label T-F units, this method mainly increases HIT rates and hence improves HIT$-$FA rates. Although rethresholding introduces some scattered interference-dominant units, it is easy to remove these units by auditory segmentation. Note that the setting of thresholds is application-dependent. In this study, we find that a small validation set is sufficient to find appropriate thresholds and they are robust to the choice of validation set. Noth that, although one can apply rethresholding to GMM based classification, the performance is still lower than that of SVM in our experiment.

Feature extraction plays an important role in classification. Pitch offers a major cue to separated voiced speech from other sounds. However, determination of pitch in noisy conditions is a difficult task. Although we can use the ground-truth pitch to generate pitch-based features in the training phase, we have to estimate pitch from mixtures in the test phase. We have tried to use the same pitch tracker to estimate pitch in both training and test phases, which generates matched features in the training and test phases. However, the models trained using the estimated pitch do not perform better than those using the ground-truth pitch extracted from clean speech. A pitch tracker has important influence on classification results. With better pitch estimation, our system should perform even better.

AMS features are easy to extract and exist in both voiced and unvoiced speech. As indicated in the results of Section 2.6, the generalizability of AMS features appears not as good as pitch-based features. Another limitation of AMS features is that they can only address nonspeech interference. For mixtures of two voices, AMS features are not able to distinguish them, but with multipitch tracking pitch-based features are still discriminative even though this chapter does not deal with separation of two voices. The combination of two types of features constitutes a complementary feature set, which performs better than either type alone. In addition, as the extracted features capture speech characteristics rather than speaker characteristics, the system is speaker-independent as shown in the Section 2.6.1.

In summary, we approach monaural speech separation as binary classification. Our system extracts pitch-based and AMS features from T-F units and utilizes SVMs to classify them. An auditory segmentation stage further improves classification results. Systematic evaluations show that our system yields accurate classification results. As demonstrated in [80] and [70], HIT$-$FA rates are correlated with speech intelligibility. Since our system achieves higher HIT$-$FA rates than Kim *et al.*'s system, it seems reasonable to expect that our system can lead to improved intelligibility.

CHAPTER 3

ON GENERALIZATION OF CLASSIFICATION BASED SPEECH SEPARATION

In the previous chapter, we have proposed to use a supervised learning approach to speech separation. However, in a supervised learning framework, if the distribution of the training set does not match that of the test set, the test performance of the trained model is not guaranteed. This chapter presents methods that require only a small training corpus and can generalize to unseen conditions. The system utilizes support vector machines to learn classification cues and then employs a rethresholding technique to estimate the IBM. A distribution fitting method is used to generalize to unseen signal-to-noise ratio conditions and voice activity detection based adaptation is used to generalize to unseen noise conditions. Systematic evaluation and comparison show that the proposed approach produces high quality IBM estimates under unseen conditions. The work presented in this chapter has been published in the *Proceedings of the 2012 IEEE International Conference on Acoustic, Speech, and Signal Processing* [38] and *IEEE Transactions on Audio, Speech, and Language Processing* [40].

3.1 Introduction

Speech communication usually takes place in complex acoustic environments. In the previous chapter, we have described a supervised learning approach to estimate the IBM for speech separation. However, for supervised learning to be effective, the distribution of the training set needs to match that of the test set. For speech separation, if input SNRs or background noises in test mixtures are not seen in the training set, the trained classifier will unlikely achieve good classification results. Hence, it is important to investigate the generalization capability of such classifiers.

In this chapter, we propose an approach to estimate the IBM under unseen SNR or noise conditions. The proposed approach consists of an SVM training stage followed by a rethresholding step. We utilize SVMs to produce initial classification boundaries and then derive new thresholds to classify T-F units in unseen acoustic environments. The new thresholds are adaptively computed based on the characteristics of test mixtures, and they are expected to generalize to new SNR or noise conditions. For unseen SNRs, by analyzing statistical properties, we determine the new thresholds by fitting the distribution of SVM outputs. For unseen noises, a voice activity detector is incorporated to construct a development set and then derive the thresholds.

The chapter is organized as follows. In the next section, we present an overview of the proposed system. Sections 3.3 and 3.4 describe how to generalize the SVM system to unseen SNR and noise conditions, respectively. Systematic evaluation and comparison are given in Section 3.5. We discuss related issues and conclude the chapter in Section 3.6.

3.2 System overview

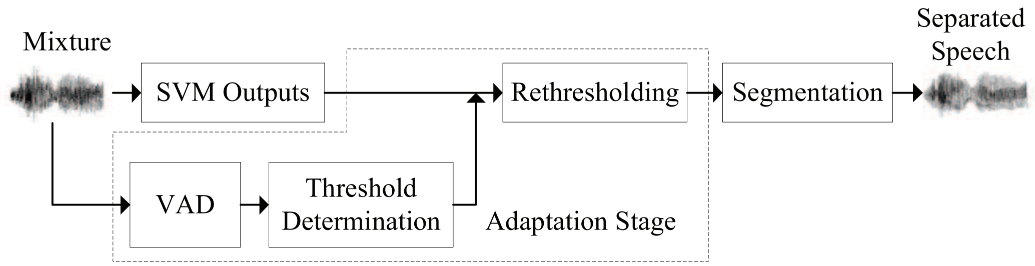


Figure 3.1: Diagram of the proposed system.

Figure 3.1 shows the diagram of the proposed system, which consists of a training phase and a test phase. In the training phase, the speech and the noise are used to create the IBM, which provides the desired output for training. The features in each T-F unit are extracted from the mixture and then used to train an SVM model in each frequency channel. In the test phase, we first use the trained SVM to initially classify T-F units, and then utilize a rethresholding technique to generalize the system under different test conditions. Auditory segmentation is used to further improve the estimated mask and separated speech is finally resynthesized by using the estimated IBM.

3.2.1 Feature extraction

An input mixture $s(t)$ is first fed into a 64-channel gammatone filterbank whose center frequencies are distributed from 50 Hz to 8000 Hz [134]. This filterbank is derived from psychophysical studies of auditory periphery and is a standard model of cochlear filtering [99]. In each channel, the output is windowed into 20-ms time frames with 10-ms frame shift, forming a cochleagram. We use $u_{c,m}$ to denote a T-F unit in the cochleagram, which corresponds to frequency channel c and time frame m.

Given the cochleagram of the mixture, we extract acoustic features from each T-F unit. In [36], a combination of pitch-based features and AMS features [126] is used to effectively classify T-F units under the noise matched condition. For SNR generalization, since we only consider the matched noises, it is reasonable to adopt the same combined features into our system.

For noise generalization, AMS features may not be an appropriate choice, because they do not show good performance under unseen noise conditions [36], [137]. According to a recent comparison of features [137], we use relative spectral transform-perceptual linear prediction (RASTA-PLP) features [46] to perform classification under unseen noise. With the pitch based features, the combined features are expected to perform good discriminative capacity on various noises.

Delta features are found to be helpful in speech separation as they encode feature variations [70], [36]. We concatenate the original features with their time delta features and frequency delta features into a combined feature vector for classification. In Section 3.5, we discuss feature extraction in details.

3.2.2 SVM and rethresholding

Similar to Chapter 2, we use SVM to classify T-F units to target-dominant or interference-dominant classes. In order to facilitate the rethresholding stage, we use probabilistic SVMs to model the posterior probability that a T-F unit label Y is assigned 1 given the feature vector, denoted as $P(Y = 1|\mathbf{x})$. A separate SVM is trained for each frequency channel because the characteristics of the speech signal in different channels can be very different. In the training phase, we use the radial basis function kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$ and the parameters are chosen by 5-fold cross-validation. To obtain a probabilistic representation, we use a sigmoid function to map an SVM decision value to a number between 0 and 1, which is then interpreted as the posterior probability of the target [100]:

$$P(Y = 1|\mathbf{x}) = \frac{1}{1 + \exp\left(\alpha f(\mathbf{x}) + \beta\right)} \tag{3.1}$$

where the parameters α and β denote the shape of the sigmoid function, which are fit using maximal likelihood estimation in the training phase. To fit the sigmoid function, we first define a new training set (f_i, t_i), where t_i is the target probability defined by the target label y_i:

$$t_i = \frac{y_i + 1}{2} \tag{3.2}$$

The parameters α and β are estimated by minimizing a cross-entry loss function:

$$\mathcal{L} = -\sum_i t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \tag{3.3}$$

where

$$p_i = \frac{1}{1 + \exp(\alpha f_i + \beta)} \tag{3.4}$$

The SVM library LIBSVM [16] is used in our experiments to estimate the parameters and produces probability outputs. With the compact representation, one can derive new thresholds within $[0, 1]$ instead of $[-\infty, \infty]$.

In the test phase, the decision value for each T-F unit is calculated from the discriminant function as follow:

$$f(\mathbf{x}) = \sum_{i \in \mathrm{SV}} a_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{3.5}$$

where SV denotes the set of support vector indices in training data and y_i is the label corresponding to $\mathbf{x}_i$. a_i is a Lagrange multiplier and b is the bias, both of which can be determined in the training phase. The decision value $f(\mathbf{x})$ is a real number between $(-\infty, +\infty)$, which is then mapped to a number within $[0, 1]$ representing the posterior probability of the unit being target-dominant using Eq. (3.1). Note that Eq. (3.1) is a monotonic bijective function but the original threshold $f(\mathbf{x}) = 0$ does not necessarily correspond to $P(Y = 1|\mathbf{x}) = 0.5$.

Generally speaking, standard probabilistic SVMs use $P = 0.5$ as the threshold to perform classification. In this study we train with a fixed input SNR or using a small number of noise types and wish to generalize to a variety of unseen conditions. In this case, we do not expect the trained SVMs to produce good classification results in unseen conditions.

In [36], we proposed a rethresholding technique to improve SVM classification results, which has been successfully used for text classification [13], [120]. One reason for the use of rethresholding is that there exists a mismatch between the training set and the test set. Under unmatched conditions, the optimal hyperplane for the training set likely deviates from the optimal hyperplane for the test set. In this study, we propose to use rethresholding to adjust trained hyperplanes in order to generalize to unseen SNR or noise conditions.

Specifically, we first need to find a channel-specific threshold θ_c that maximizes the classification accuracy in channel c, and then use the new threshold to binarize $P(Y = 1|\mathbf{x})$:

$$Y = \begin{cases} 1, & \text{if } P(Y = 1|\mathbf{x}) > \theta_c \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

Our experiments show that with properly chosen thresholds, the system can significantly improve classification. For SNR generalization, the system is trained on 0 dB. Therefore, the key problem is how to determine the new threshold θ_c for each channel c. In [36], a small validation set is used to determine the thresholds. But this strategy cannot be used in this study, because the statistical properties of the test set are very different from those of the training set and unknown. Thus, we need to develop new strategies for rethresholding under unseen SNR or noise conditions, which are described in detail in Section 3.3 and Section 3.4, respectively.

3.2.3 Auditory segmentation

The rethresholded mask gives a good estimate of the IBM, but it still misses some target-dominant units and contains some interference-dominant units. To further improve the rethresholded mask we utilize auditory segmentation which takes into consideration contextual information beyond individual T-F units. We adopt the same segmentation method as in 2.5.

To summarize, given a noisy speech signal, we first extract features in the T-F domain and then use SVM to produce initial classification for each T-F unit. Then, we use rethretholding to adapt SVM output under different conditions. Finally, auditory segmentation is used to improve the estimated IBM. The following sections describe how to apply rethresholding under different conditions.

3.3 Generalization to different input SNRs

For SNR generalization, the training set contains mixtures at a single input SNR and the system will be tested on mixtures at different input SNRs. In this case, if we directly use $\theta = 0.5$ as the threshold, the system does not generalize well to unseen SNRs. We refer to the threshold that maximizes some classification accuracy as the optimal threshold. We observe that in unmatched SNR conditions, the use of the optimal threshold in each channel can substantially improve the classification result relative to the default threshold of 0.5. In other words, if we can find thresholds close to the optimal one, the system is expected to generalize well under unseen SNR conditions.

Furthermore, we observe that, although the optimal threshold varies in different SNR conditions, SVM outputs have similar distribution shapes and the optimal thresholds are located at similar positions relative to the distribution shapes. As a typical example, Fig. 3.2 shows the histograms of the SVM outputs in the 18th channel for a female utterance, "A man in a blue sweater sat at the desk," from the IEEE corpus [110] mixed with speech-shaped noise. The system is trained on 100 IEEE sentences mixed with speech-shaped noise, factory noise and babble noise at 0 dB and SVM outputs are generated at -10, -5, 0, 5 and 10 dB input SNRs. The figure shows that there exists a peak K on the left side ($P < 0.6$) of each histogram. Also, the SVM ouputs on the left side for different SNRs have similar distribution shapes which gradually become sharper as the input SNR increases. Further, the optimal threshold θ shown as the solid vertical line in each histogram is increasingly close to the peak K as the distribution becomes shaper. If we only consider the SVM outputs on the left side of the histogram, the optimal threshold always occurs at the tail end of the distribution under each input SNR condition. This motivates us to use the same distribution function to fit SVM outputs at different SNRs with different parameter values.

One can perform distribution fitting in two ways: fit all SVM outputs less than 0.6 or SVM outputs between K and 0.6. We have explored several reasonable distributions as the candidates to fit SVM outputs and use the Kolmogorov-Smirnov

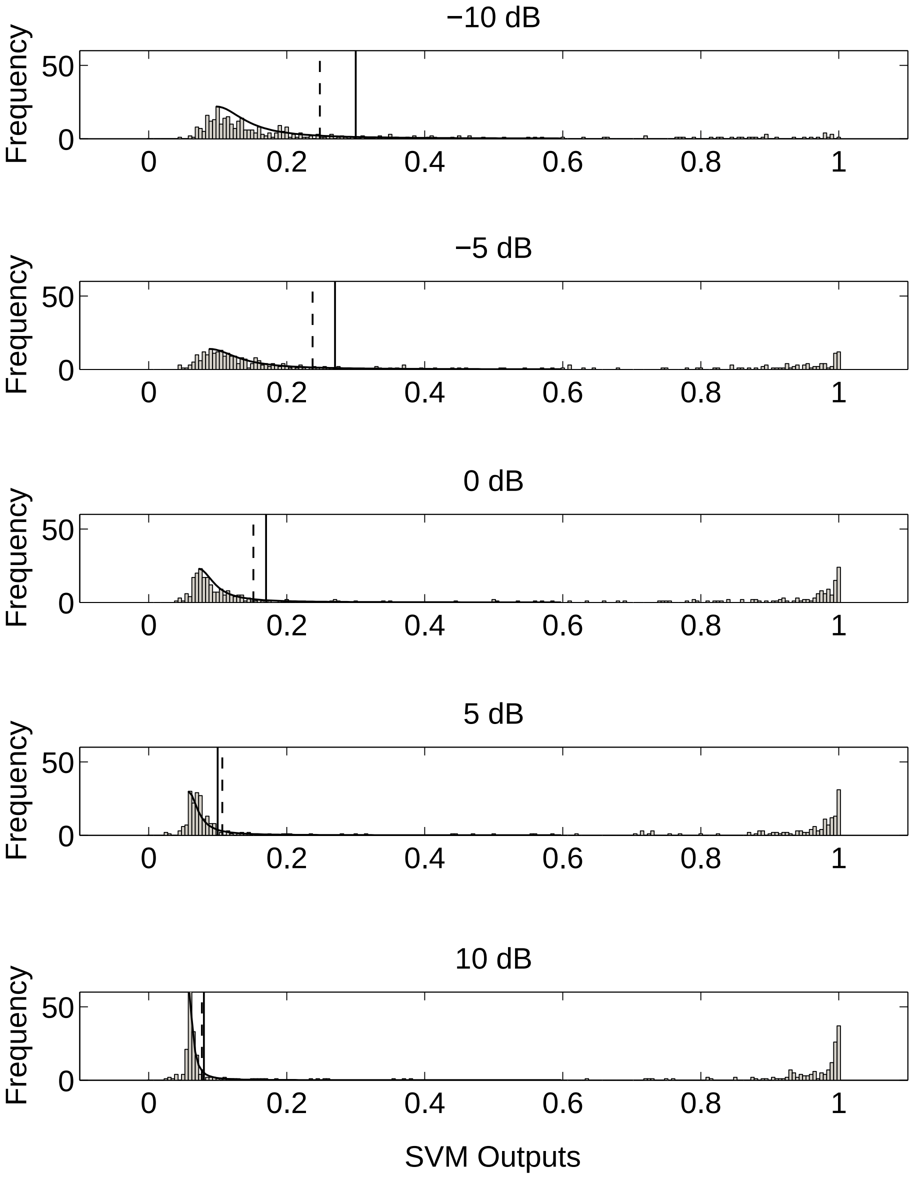


Figure 3.2: Histograms of the SVM outputs in the 18th channel with different input SNRs. The solid curve denotes the half-Cauchy distribution used to fit the SVM outputs. A vertical line indicates the optimal threshold and a dashed vertical line the estimated optimal threshold by using distribution fitting.

(K-S) statistics [11] to test the goodness of fit. Three distributions are tested: the generalized extreme value distribution (GEV) is used to fit all SVM outputs less than 0.6, whereas the half-Cauchy and the half-Laplace distributions are used to fit the SVM outputs within $[K, 0.6]$. The probability density functions are:

$$\text{GEV:} \quad f(x;\mu,\sigma,\xi) = \frac{1}{\sigma}[1+\xi(\frac{x-\mu}{\sigma})]^{\frac{-1}{\xi}-1}\exp\{-[1+\xi(\frac{x-\mu}{\sigma})]^{-\frac{1}{\xi}}\} \tag{3.7}$$

$$\text{Half-Cauchy:} \quad f(x;\mu,\sigma) = \begin{cases} \dfrac{2}{\pi\sigma[1+(\frac{x-\mu}{\sigma})^2]}, \text{if } x \geq \mu \\ 0, \text{ otherwise} \end{cases} \tag{3.8}$$

$$\text{Half-Laplace:} \quad f(x;\mu,\sigma) = \begin{cases} \dfrac{1}{\sigma}\exp{(-\dfrac{x-\mu}{\sigma})}, \text{if } x \geq \mu \\ 0, \text{ otherwise} \end{cases} \tag{3.9}$$

where, μ, σ, ξ are parameters determined by maximal likelihood estimation.

Fig. 3.3 shows the K-S statistic test results in each channel, averaging over 10 IEEE sentences mixed with the three noises at five SNR levels. From the figure, all three distributions achieve relatively low K-S statistics, meaning that the candidate distributions fit the data well. The best one is the half-Cauchy distribution which has the lowest K-S statistics in most channels. Consequently, we use the half-Cauchy distribution in our method. As shown in Fig. 3.2, under each SNR condition, the solid curve denotes the probability density function of a half-Cauchy distribution, which well fits the SVM outputs between K and 0.6.

Therefore, given SVM outputs in one channel, we estimate parameters of a half-Cauchy distribution to fit the outputs by maximal likelihood estimation. Based on the fitted distribution function F and the observation that the optimal threshold

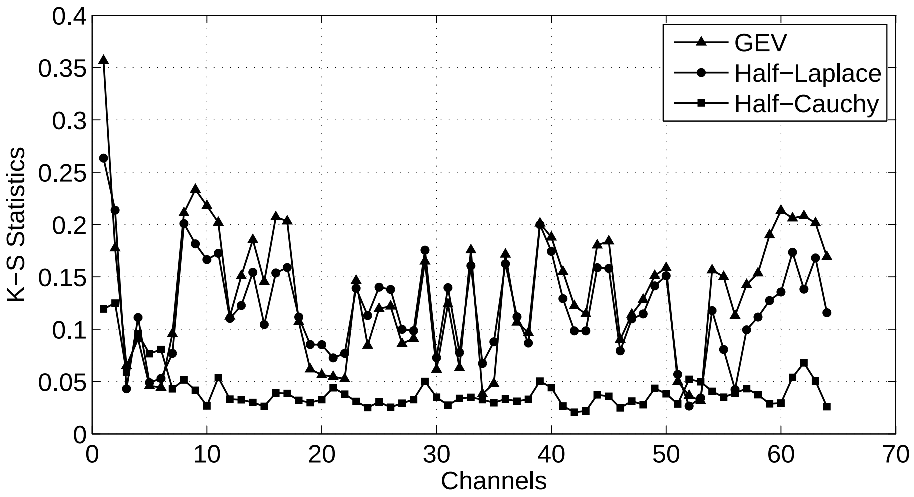


Figure 3.3: Kolmogorov-Smirnov statistics for three distributions

θ is located at the tail end of the distribution, the corresponding cumulative probability $\rho = F(\theta)$ should be close to 1. This turns optimal threshold estimation to another problem: given F with unknown parameters $\Omega = \{\mu, \sigma\}$ and a predetermined cumulative probability ρ, we can first estimate Ω based on SVM outputs and then approximate the optimal threshold by calculating the inverse cumulative distribution function $\theta = F^{-1}(\rho; \Omega)$. Here, ρ is set to 0.9, which is chosen from a validation set. Incidentally, we choose 0.6 instead of 0.5 as the upper bound of the SVM outputs to fit the distribution because we want to include more samples for the fitting. The number of SVM outputs less than 0.6 is very unlikely too small (i.e., less than 5% of the total SVM outputs) to well fit a distribution function, because human speech contains pauses which should produce sufficient interference-dominant units (i.e., SVM outputs less than 0.6) in the mixture. In the case that few SVM outputs can be used to fit the distribution, we simply set the threshold to the original value 0.5. Note that, although we only use those SVM outputs less than 0.6 to fit a distribution function, it does not mean that 0.6 is the upper bound of the estimated threshold. The threshold only depends on the parameters of the fitted distribution function and can be any value in $[0, 1]$.

To summarize, we use the following algorithm to estimate the optimal threshold θ in each channel:

1. Given the SVM outputs, we uniformly divide $[0, 1]$ into 100 bins and derive the histogram of SVM outputs. For those bins less than 0.6, we choose the bin with the highest frequency as the peak K.

2. We use the half-Cauchy distribution F with unknown parameters Ω to fit the SVM outputs within $[K, 0.6]$ and use maximal likelihood to estimate Ω;

3. We estimate the optimal threshold using inverse cumulative distribution function $\theta = F^{-1}(\rho; \Omega)$.

The dashed line shown for each histogram in Fig. 3.2 denotes the estimated optimal threshold based on distribution fitting, which is close to the optimal threshold. Finally, we use the threshold calculated from the algorithm to binarize the SVM outputs in each channel and obtain a rethresholded mask. This mask is further improved by an auditory segmentation procedure and form an estimated IBM. It is worth emphasizing that this method estimates optimal thresholds only based on SVM outputs of the mixture without the knowledge of the input SNR.

3.4 Generalization to different noises

Another important issue is generalization to unseen noises. We also use rethresholding to generalize the system to unseen noises as we have observed that optimal thresholds significantly improve the classification results.

Although distribution fitting is able to generalize the trained models to unseen SNR conditions, it does not work well for unseen noise conditions because the characteristics of noises can be very different and no pattern of the histograms appears to fit all noises. Fig. 3.4 shows histograms of SVM outputs in the 18th channel corresponding to four female utterances mixed at 0 dB with (a) speech-shaped noise and (b) rock music. Both noises are not seen in the trained SVM model (see Section 3.5.2 for more details). The solid vertical line indicates the optimal threshold in each histogram. We can compare the histograms in (a) and (b) in each row. Although the same sentence is used to generate the SVM outputs, they have very different distributions as the noises are different. On the other hand, for those mixtures with the same type of noise, the optimal thresholds have close values: around 0.5 for speech-shaped noise and 0.8 for rock music. The histograms for speech-shaped noise in this figure are quite different from those in Fig. 3.2 for two reasons: (1) speech-shaped noise is contained in the training set in Fig. 3.2 but not in the training set in Fig. 3.4; (2) the system in Fig. 3.2 uses AMS and pitch-based features for SNR generalization but the system in Fig. 3.4 uses RASTA-PLP and pitch-based features for noise generalization.

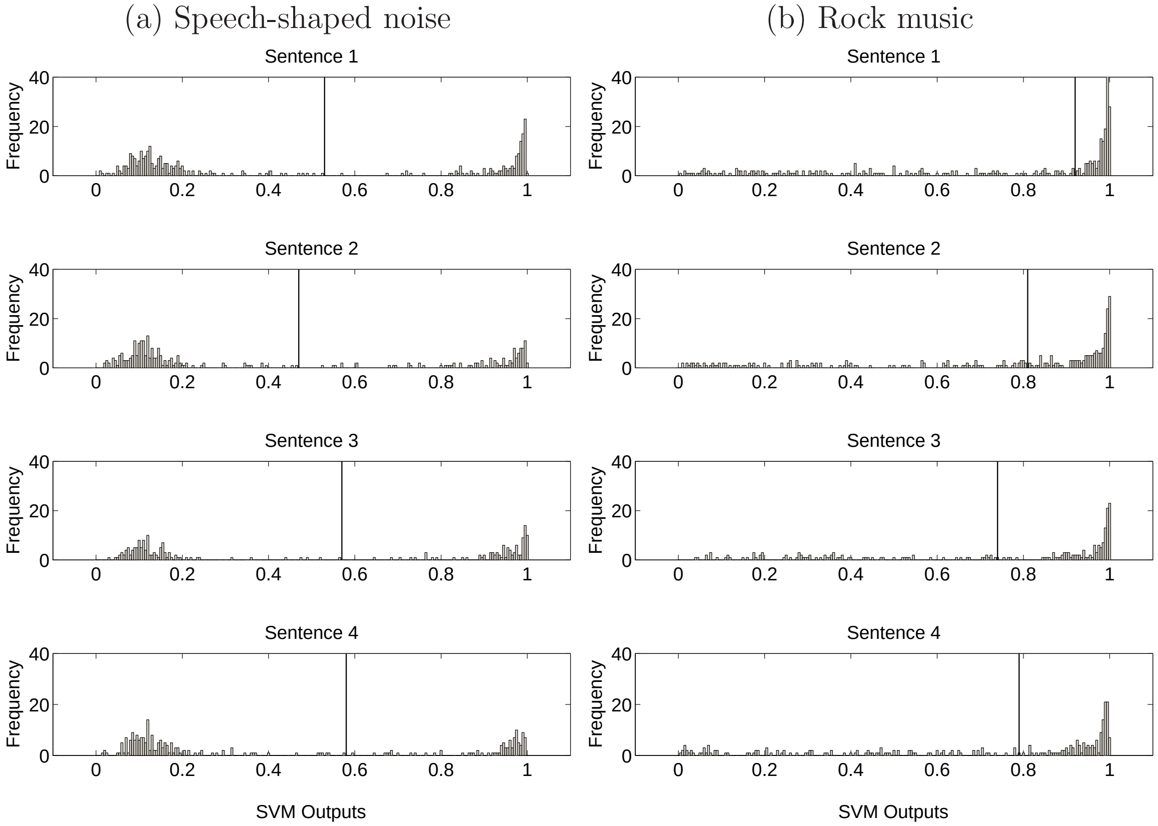


Figure 3.4: Histograms of the SVM outputs in the 18th channel. Four different utterances mixed with (a) speech-shaped noise and (b) rock music. The solid vertical line in each panel denotes the optimal threshold.

The above analysis suggests that, if mixtures come from the same kind of noise, it is reasonable to apply the same threshold to all these mixtures in each channel. In other words, although it is impossible to directly obtain the optimal thresholds for a test mixture as the IBM is not accessible, if we can somehow access part of the noise, we can use the noise part to construct a development set including a reference mixture and the corresponding IBM to calculate the optimal thresholds. The optimal thresholds obtained from the development set are expected to perform well on the test mixture because the same type of noise is used in both mixtures. Obviously, to construct the development set clean speech is needed, which can be an arbitrary utterance. We randomly choose a single utterance, “Shake the dust from your shoes, stranger,” from the IEEE corpus and use this one to construct the development sets for all test mixtures.

To obtain noise portions from a test mixture, we propose to apply voice activity detection (VAD) in an adaptation stage to perform rethresholding. VAD is used to identify noise-only frames which are then mixed with the above clean speech to construct a development set. The thresholds chosen from the development set are used to produce a binary mask. Fig. 3.5 illustrates the computational flow.

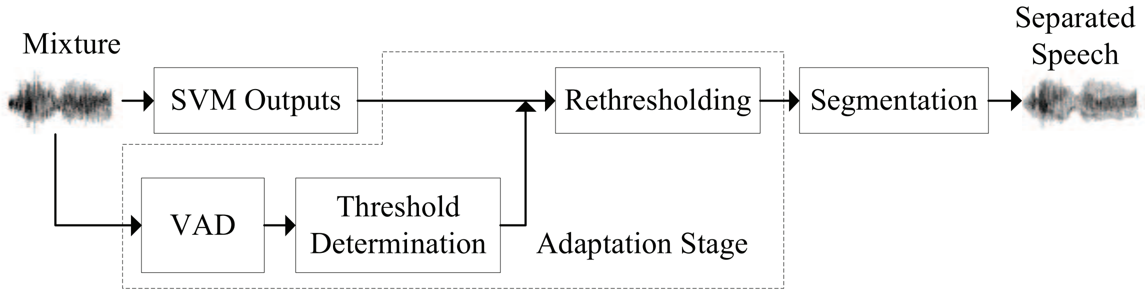


Figure 3.5: Diagram of VAD based rethresholding for generalization to unseen noises.

As shown in the figure, given a test mixture, we use the trained SVMs to output the posterior probability of speech dominance for each T-F unit. In parallel, we use Sohn *et al.*’s VAD algorithm [119] to detect noise frames. This standard VAD algorithm uses a statistical model-based method to produce the likelihood of speech presence for each frame. In our corpus, speech pause accounts for around 30% of frames, so we select 30% of the frames with the lowest likelihoods as the candidates of noise frames. To avoid spurious noise frames caused by VAD errors, we further use detected pitch in the feature extraction stage in Section 3.2.1 to improve the VAD results: a candidate of noise frame is removed if a pitch is detected in this frame. In addition, since very short noise sections are not useful for constructing a development set, we exclude those noise sections whose lengths are shorter than 50 ms (or 5 frames).

With detected noise frames and the one clean utterance, we mix them into a reference mixture. This mixing, however, requires that the noise frames and the clean utterance have the same length. In this study, both the test mixture and the clean utterance last around 2 seconds, and as a result the total length of detected noise frames is usually significantly shorter than the length of the clean utterance. To match the utterance length, we first concatenate detected noise frames to a noise section and then repeatedly duplicate the noise section until the total length is equal to that of the utterance. The resulting noise section and the clean utterance are used to construct a development set. We find that, although a longer test mixture (> 10 seconds) can provide more noise frames without duplication, it does not give better results than a 2-second mixture.

After we construct a development set containing a single mixture, we calculate optimal thresholds θ based on the reference mixture and its IBM. That is, we apply our trained models to the reference mixture to calculate SVM outputs and use the corresponding IBM to choose the optimal threshold θ in terms of accuracy in each frequency channel. With the obtained θ and SVM outputs of the test mixture, it is straightforward to use Eq. (3.6) to produce a rethresholded mask. Finally, we employ a segmentation step to further improve IBM estimation.

3.5 Evaluation and comparison

3.5.1 Generalization results for unseen SNRs

We first evaluate the capacity of our system to generalize to unseen SNRs. As we mentioned above, we utilize pitch-based features, AMS features and their delta features for SNR generalization. For pitch-based features, we calculate the normalized autocorrelation function $A(c, m, \tau)$ at pitch period $\tau_S(m)$. For voiced speech, $A(c, m, \tau_S(m))$ measures how well the unit response is consistent with the target pitch, which has been proven to be an effective feature for speech separation [54], [63]. To remove the influence of pitch errors in the training phase, we use *Praat* [10] to extract the ground-truth pitch from the premixed speech in the training phase, and use the pitch tracker in [65] to extract the estimated pitch from the mixture in the test phase. Similarly, we also compute autocorrelation from the envelope of the response to obtain $A_E(c, m, \tau_S(m))$ as a feature to capture amplitude modulation information.

We calculate delta features in the following manner: in the time dimension, for $m \geq 2$, the time delta feature $\Delta A^M(c, m, \tau_S(m)) = A(c, m, \tau_S(m)) - A(c, m - 1, \tau_S(m))$; $\Delta A^M(c, 1, \tau_S(m))$ is simply set to $\Delta A^M(c, 2, \tau_S(m))$ for convenience. We compute the frequency delta feature $\Delta A^C(c, m, \tau_S(m))$ in the same way. Therefore, between response and envelope autocorrelation we get a 6-dimensional pitch-based features $\mathbf{x}_P$.

We use the same method as in [36] to extract AMS features. Specifically, the envelope from the filter response within each T-F unit is extracted. The envelope is Hanning windowed and zero-padded for a 256-point FFT. The resulting FFT magnitudes are integrated by 15 triangular windows, generating a 15-dimensional AMS feature. Similarly, we calculate delta features across time frames and frequency channels. In each T-F unit, the pitch-based feature vector $\mathbf{x}_P$ and the AMS feature vector $\mathbf{x}_A$ are combined into a feature vector and used for the classification under different SNR conditions.

The features are extracted from the IEEE corpus [110]. Similar to Kim *et al.* [70], the training set consists of 100 female utterances mixed with three types of noise: speech-shape noise, factory noise and babble noise at 0 dB. For the test set, we choose 10 new utterances mixed with the same three types of noise at -10, -5, 0, 5 and 10 dB.

In order to quantify the performance of our system, we compute the HIT rate which is the percent of the target-dominant units in the IBM correctly classified, and the false-alarm (FA) rate which is the percent of the interference-dominant units in the IBM wrongly classified. We use the difference between HIT and FA, HIT−FA, as an evaluation criterion since it has been shown to be correlated to human speech intelligibility [80], [70] and has been adopted in earlier studies [70], [36].

Fig. 3.6 shows the average HIT−FA results over the three noises under each input SNR condition. The triangle line indicates the original HIT−FA rates without rethresholding. With the optimal thresholds, the HIT−FA rates are boosted by 10% absolute on average, which clearly shows the advantage of rethresholding. By using distribution fitting based rethresholding, we improve the HIT−FA results by 9% for low input SNR conditions (-10 and -5 dB) and 10% for high SNR conditions (5 and 10 dB). The result in the matched SNR condition is also improved, probably because the ground-truth pitch is used in the training phase but the estimated pitch is used in the test phase. This pitch discrepancy would lead to an optimal threshold different from the original threshold 0.5 (as shown in Fig. 3.2), so the HIT−FA rate could be improved by rethresholding even under matched SNR conditions. No segmentation is used in this comparison. It is interesting to note that, the distribution fitting based rethresholding outperforms the optimal rethresholding under the -10 dB condition. This is because the optimal threshold is chosen to maximize the accuracy in each channel, which does not necessarily maximize the corresponding HIT−FA rate for the whole mask (see [36]).

The above results show the advantage of rethresholding in our system. We now compare our system with three recent speech separation systems. The first one is an IBM estimation system proposed by Kim *et al.* [70]. As mentioned in Section 3.1, this system extracts AMS features and utilizes GMM classifiers to estimate the IBM, and it has been demonstrated to improve speech intelligibility in human listening tests. Their system is trained on the same 100 utterances mixed with the same three

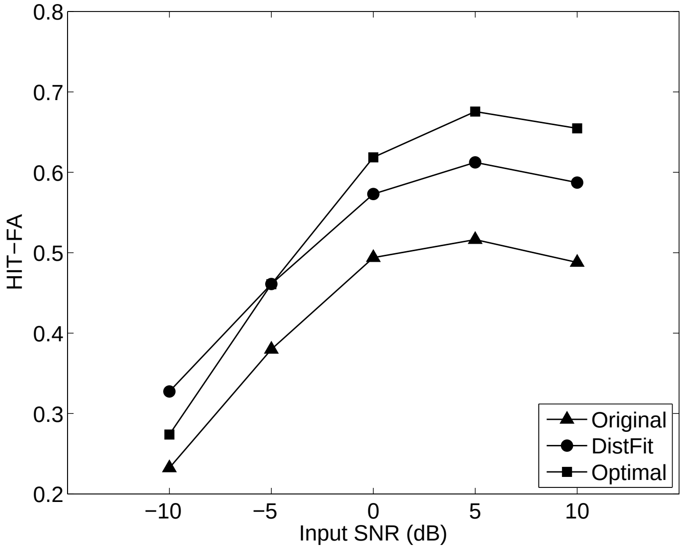


Figure 3.6: Distribution fitting based SNR generalization results in terms of HIT−FA. The line with triangles denotes the original SVM results, the line with circles the distribution fitting based rethresholding results, and the line with squares the results using optimal thresholds.

noises, but three SNR levels of at -5, 0 and 5 dB SNR as reported in [70]. We train a 256-component GMM for each class in each channel. The second one is a state-of-art speech enhancement system based on noise tracking proposed by Hendriks *et al.* [45]. This system assumes that both the speech and noise DFT coefficients have a complex-Gaussian distribution and utilizes an MMSE estimator of the noise magnitude-squared DFT coefficients to estimate noise power spectral density. The clean speech DFT coefficients are estimated from a magnitude-DFT MMSE estimator presented in [28]. With these estimates, one can calculate the speech and noise energy within a time-frequency unit in the linear DFT domain. Since our IBM is defined in the gammatone filterbank domain, we need to convert the speech and noise energy in the linear DFT domain to the corresponding energy estimates in the gammatone filterbank domain [92]. Without loss of generality, we consider the energy E of a T-F unit $u_{c,m}$ in the gammatone filterbank domain:

$$\begin{aligned} E(c,m) &= \sum_{n} |y_c[n]|^2 = \frac{1}{K}\sum_{k=0}^{K-1} |Y_c[k]|^2 \\ &= \frac{1}{K}\sum_{k=0}^{K-1} |X[k]|^2 \cdot |G_c(k)|^2 \end{aligned} \tag{3.10}$$

where $y_c[n]$ denotes a filtered time domain signal in frequency channel c and frame m, and $Y_c[k]$ are the DFT coefficients of $y_c[n]$, where K is set to 512 in our experiments. The second equation is due to Parseval's theorem [95]. G_c is the frequency response function of the gammatone filter in channel c. $X[k]$ is a DFT coefficient of the original signal, which can be estimated by Hendriks *et al.*'s system. For each T-F unit in the gammatone filterbank domain, we use Eq. (3.10) to calculate the speech and noise energy respectively, and then compute the local SNR to generate the binary mask. The third method is a model-based system using a general framework proposed by Ozerov *et al.* [97]. This method utilizes NMF to perform separation. We use 10 IEEE sentences to train a 64-component speaker NMF model and the same three noises to train a 16-component noise NMF models. Since the NMF-based method produces the separated speech signal and noise signal in the time domain directly, we decompose these two signals to the T-F domain and calculate local SNRs to form a binary mask for comparison.

As shown in Fig. 3.7, the proposed system slightly outperforms the NMF-based method (by around 4% on average) in terms of HIT$-$FA rates. The other two systems

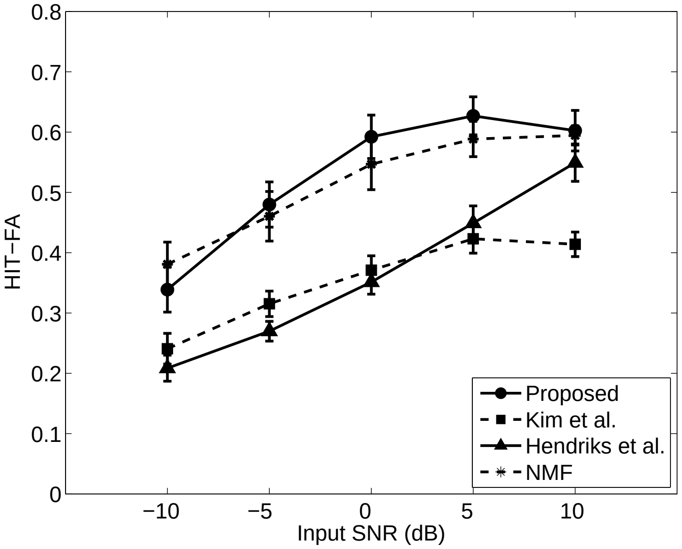


Figure 3.7: HIT−FA rates with respect to input SNR levels. The error bars indicate 95% confidence intervals of the means.

perform considerably worse. To indicate statistical significance, we also show 95% confidence intervals in the figure, which are calculated from a normal distribution fitted by obtained results. Note that, Kim *et al.*'s system is trained on -5, 0 and 5 dB input SNRs, and it is supposed to achieve good performance at the three trained input SNRs.

We should point out that Hendriks *et al.*'s system is not designed to estimate the IBM. We have also implemented a binary masking system proposed by Jensen and Hendriks [62] for comparison. Their system derives a gain function based on the same spectral magnitude MMSE as in Hendriks *et al.* but generates an optimal binary mask in the MMSE sense, which is a binarization based on gain thresholds. We first calculate a gain threshold for each T-F unit and convert it to an energy threshold in the DFT domain. Eq. (3.10) is then used to calculate the corresponding energy threshold for each T-F unit in the gammatone filterbank domain. With the cochleagram of the mixture and the calculated energy thresholds, we can generate an optimal binary mask in the gammatone filterbank domain. However, their system achieves lower HIT−FA rates than the one based on Hendriks *et al.* described above. One important reason is that Jensen and Hendriks aim to obtain the optimal binary mask in the MMSE sense rather than the ideal binary mask used in our study. This suggests that there are differences between an optimal-binary-mask estimator and an ideal-binary-mask estimator. Even though Jensen and Hendriks [62] reported that estimated optimal binary masks do not lead to significant improvements of speech intelligibility, the same cannot be said of estimated IBMs [70].

The comparisons above focus on unit classification accuracy, where we need to convert the energy estimates from Hendriks *et al.* in the DFT domain and the separated signals from the NMF-based method in the time domain to the gammatone filterbank domain. To eliminate the effects of conversion, we use inverse FFT to resynthesize estimated speech energy in the DFT domain to the waveform. We also resynthesize from the estimated IBMs of Kim *et al.* and the proposed system to waveform [134]. With the resynthesized signal, we measure the output SNR of the separated speech as follows [54]:

$$\text{SNR} = 10 \log_{10} \frac{\sum_n s_I^2(n)}{\sum_n [s_I(n) - s_E(n)]^2} \tag{3.11}$$

For Kim *et al.* and the proposed system, $s_I(n)$ and $s_E(n)$ indicate the signals

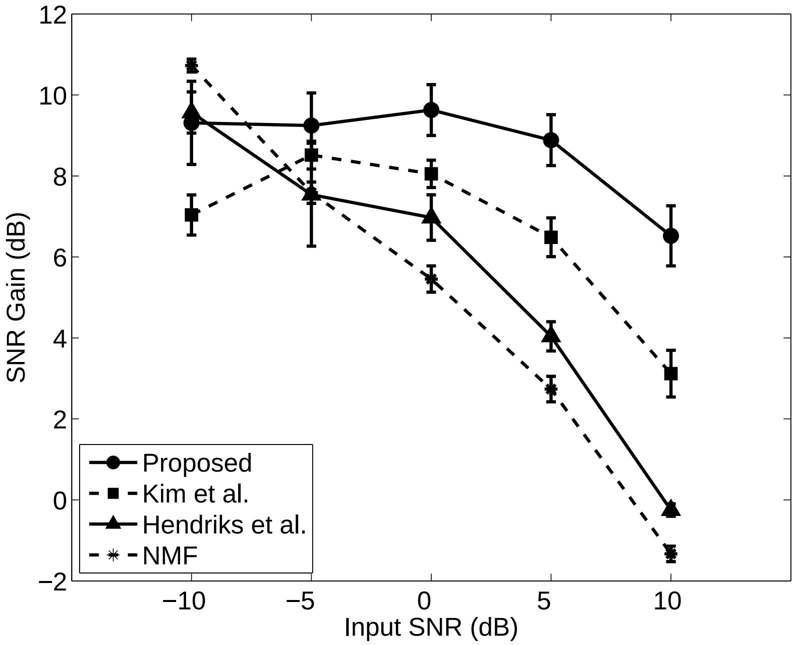


Figure 3.8: SNR gains with respect to input SNR levels. The error bars indicate 95% confidence intervals of the means.

resynthesized using the IBM and the estimated IBM, respectively. For Hendriks *et al.*'s system, $s_I(n)$ and $s_E(n)$ indicate the clean speech and the signal resynthesized using the estimated speech energy, respectively. For the NMF-based method, $s_I(n)$ and $s_E(n)$ indicate the clean speech and the separated speech signal, respectively. To quantitatively evaluate the performance, an SNR gain is computed by subtracting the output SNR of separated speech by the input SNR before separation. Fig. 3.8 shows the SNR gains. The proposed system achieves considerable SNR gains at all input SNRs. Although the SNR gains of all systems decrease gradually as the input SNR increases, the other three systems have more significant degradation at higher input SNRs.

3.5.2 Generalization results for unseen noises

We utilize pitch-based features, RASTA-PLP features and their delta features for SNR generalization. To get RASTA-PLP features, after the power spectrum is warped to the Bark scale, we log-compress the resulting auditory spectrum, filter it by the RASTA filter, and expand it by an exponential function. Subsequently, PLP analysis is taken on this filtered spectrum. The original RASTA-PLP feature is a 13-dimensional vector and we also calculate the delta features for RASTA-PLP across time frames and frequency channels to generate a 39-dimensional RASTA-PLP feature vector $\mathbf{x}_R$. The pitch-based feature vector $\mathbf{x}_P$ and the RASTA-PLP feature vector $\mathbf{x}_R$ are finally combined and a 45-dimensional feature vector for each T-F unit is used as the input to the classifier for noise generalization.

To evaluate generalization to unseen noises, we choose 30 female utterances from the IEEE corpus mixed with 5 types of noise out of a 100 nonspeech noise set [52] at 0 dB SNR to train the system. To construct a representative training set, we use a clustering based noise selection scheme to choose training noises. Intuitively, we want to include the most diverse noises as the training set, i.e., the distribution of features extracted from the training noises should cover the feature space as much as possible. For noise selection, we only consider RASTA-PLP features since pitch-based features do not exist in unvoiced speech. We first pass each noise waveform through a gammatone filterbank and then extract RASTA-PLP features from each T-F unit. Then, the mean of the RASTA-PLP features is calculated over all units for each type of noise. Therefore, each noise is represented by a 13-dimensional feature vector. We then apply the K-means (K=5 in this experiment) clustering to these 100 feature vectors and thus 100 noises are divided into 5 clusters. For each cluster, we select one noise that has the shortest distance to the cluster center as the representative. Therefore, 5 representative noises are used in the training set. Compared with random noise selection, this clustering-based noise selection produces 3% improvement in terms of HIT−FA.

To test our system, we use 10 new female utterances mixed with the 10 types of noise—N1: speech-shape noise, N2: factory noise, N3: fan noise, N4: bird chirp, N5: white noise, N6: cocktail party noise, N7: rain noise, N8: rock music, N9: wind noise, N10: clock alarm—at 0 dB. The test noises cover both stationary and nonstationary noises and have very different frequency characteristics, and none of them are in the training set.

Fig. 3.9 shows the HIT−FA results of the proposed system. For each noise, the left two bars show the original SVM results using a threshold of 0.5 and the rethresholding results using the optimal thresholds, respectively. The figure shows that the optimal rethresholding substantially improves HIT−FA and achieves an average improvement of 7.3%, which suggests the utility of rethresholding for generalization. The VAD based rethresholding improves HIT−FA rates under all unseen noise conditions and the average improvement is 5.9%. With segmentation, the proposed system further improves IBM estimation, and it outperforms the original one by 7.4% making it

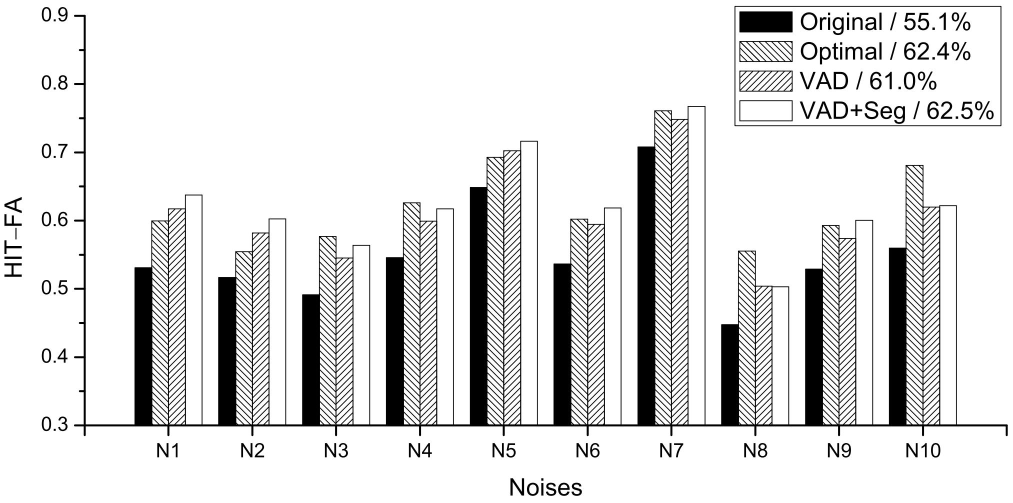


Figure 3.9: Noise generalization in terms of HIT−FA. "Original" denotes the original SVM results without rethresholding, "Optimal" the rethresholding results using optimal thresholds, and VAD denotes the VAD based rethresholding results. VAD+Seg denotes the results using VAD based rethresholding followed by segmentation.

comparable to the optimal rethresholding results. These results demonstrate that, with a little adaptation, our system generalizes well to different noise conditions.

Since our system utilizes nonspeech intervals detected by the VAD algorithm to adapt the thresholds, we also adopt a similar strategy in Ozerov *et al.* [96] for the model-based system where the noise model is adapted by the detected nonspeech intervals. In our experiment, we first train a 64-component speaker NMF model using 10 IEEE sentences (see Sect. 3.5.1). In the test phase, we use the same VAD algorithm as in the proposed system to extract noise frames from the mixture, and then use these noise frames to train a 16-component noise NMF model. Finally, we use the obtained speaker model and noise model as priors to separate speech.

In addition, we compare with the systems described in Section 3.5.1. Fig. 3.10 shows the comparative results in terms of HIT−FA rates. As shown in the figure, the proposed system achieves the highest HIT−FA rates except for N1, N5 and N9 where NMF-based system performs slightly better. On average, the proposed system outperforms the NMF-based method by around 5%, which is statistically significant from confidence intervals.

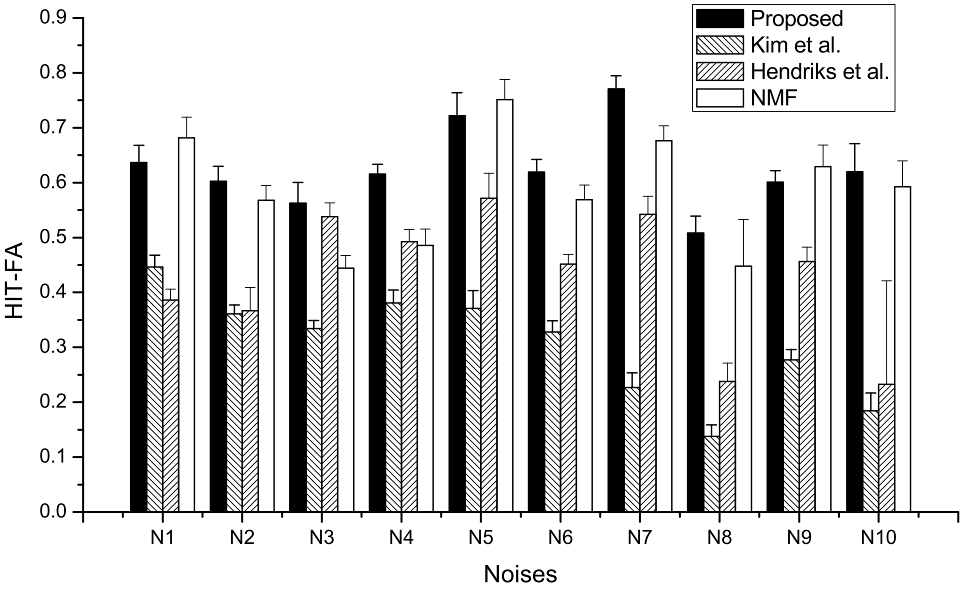


Figure 3.10: Noise generalization comparisons in terms of HIT−FA. The proposed method denotes VAD based rethresholding followed by segmentation. The error bars indicate two-side 95% confidence intervals of the means, with only one side shown for clarity.

As described in Section 3.5.1, we can resynthesize waveform signals for the four systems and calculate SNR gains. Fig. 3.11 shows such results. Our system improves SNRs by 5 dB to 12 dB, depending on noise type, and it performs better than Kim *et al.* by 3.7 dB, Hendriks *et al.* by 3.4 dB, and NMF-based system by 2.1 dB on average.

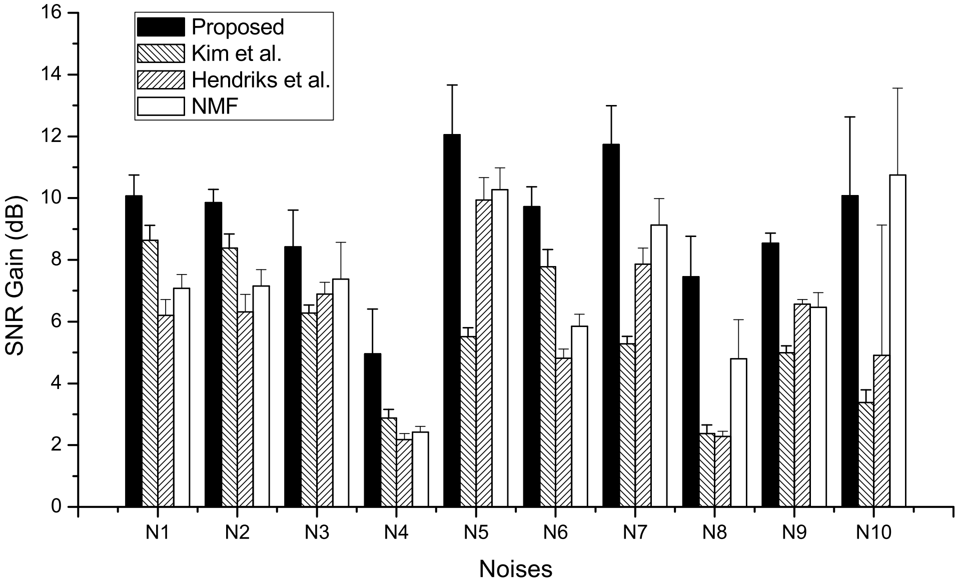


Figure 3.11: Noise generalization comparisons in terms of SNR gain.

3.6 Discussion

Monaural speech separation is a fundamental problem in speech processing. Supervised learning algorithms have been shown to be effective for speech separation, but a major issue for supervised learning is the capacity of generalization to unseen conditions, as the training set and the test set can have dissimilar properties. If this issue is not addressed, one cannot expect the trained model to perform well in unmatched conditions.

This study builds on SVM classification. An SVM outputs binary labels according to decision values, which in essence give a distance measure to the decision hyperplane, corresponding to the confidence of classification. Under many unseen conditions, the trained SVM model does not completely fail, but the optimal hyperplane just skews from the trained hyperplane to some extent. Our analysis suggests that it is possible to improve classification results by adjusting the hyperplane, which is equivalent to using a new threshold to binarize output values. Therefore, the key idea of generalization in this study is to use rethresholding to adapt the trained model to unseen conditions and the generalization issue becomes how to find appropriate thresholds. Recent research on dataset shift in classification deals with the mismatch problem between the training data and the test data [89]. In our study, shifted data lead to changes of $P(Y|\mathbf{x})$, resulting in a shift of the optimal decision boundary. In this case, rethresholding is equivalent to adjusting SVM outputs $P(Y|\mathbf{x})$. It would be interesting to explore the formulation of rethresholding as dataset shift in future work.

In this study, we convert decision values to posterior probabilities. With the probabilistic interpretation of SVM outputs, a straightforward idea to deal with generalization is to perform probabilistic inference using prior knowledge. However, too many unpredictable variables affect the probabilistic inference, and it is very difficult to directly use the Bayesian formula to derive an appropriate threshold. Instead, we use probabilities to provide initial classification and incorporate statistical properties of the test mixture to classify T-F units. Here, we prefer probabilities to decision values because the probabilistic representation provides a uniform range of $[0, 1]$ for rethresholding. We should state that rethresholding is not able to completely resolve the generalization issue, because even optimal thresholds may not be good enough, e.g., to achieve greater than 80% HIT$-$FA rates. However, as rethresholding directly focuses on the outputs of the trained model and does not require extra training, it is easy to incorporate into existing systems for improved generalization.

Under unseen SNR conditions, although the trained hyperplanes cannot be directly used to classify T-F units, the statistical properties of SVM outputs exhibit similarity at different SNRs, which provides a basis to adjust the hyperplanes. Although one can train SNR-dependent models for speech separation under different SNRs, the system would be complicated and it needs an SNR detector which is not a trivial problem. The proposed distribution fitting based rethresholding determines the thresholds only based on the test mixture and does not require any input SNR estimation.

This distribution fitting method does not work under unseen noise conditions, as no distribution is able to characterize the SVM outputs of various noises. Indeed, we tried a function approximation approach that learns a mapping from SVM outputs to optimal thresholds. However, such a mapping is not applicable to all noise types.

Instead, we use VAD to detect a small amount of noise and construct a development set to choose thresholds. Obviously, the performance of our system depends on the VAD algorithm. To improve VAD results, we utilize detected pitch to remove spurious noise frames. This strategy provides a reliable set of noise frames. This is confirmed in our experiments where clean speech, rather than noisy speech, is used to produce ideal VAD results. The experiments do not show significantly better performance by using the ideal VAD results. Therefore, our pitch-improved VAD method is not a bottleneck of the proposed system.

Obviously, features play a crucial role in classification. We use pitch-based features and AMS features for unseen SNR generalization, as this combination has proven to be effective under matched noise conditions. For noise generalization, we use pitch-based features and RASTA-PLP features, both of which capture speech information and are robust to different noise conditions. Other features may also show robust performance under different noisy conditions, but here we are only concerned with generalization based on trained classifiers and do not focus on the selection of robust features (see [137]). We point out that, since AMS features and RASTA-PLP features are not able to distinguish different voices and the VAD algorithm can only detect nonspeech intervals in a noisy mixture, our system cannot be applied to separate multiple talkers.

In this study, we address the generalization problems to different SNRs and different noises separately. In practice, both situations may need to be considered simultaneously. In such situations, rethresholding may still be applicable. Future research is required to address this more challenging case, and may involve some form of SNR detection to jump start the separation process.

To conclude, we aim to design a speech separation system that requires minimal training but is generalizable to unseen conditions. The proposed system trains SVMs to provide initial classification and then uses the rethresholding technique to estimate the IBM. To determine the thresholds under unseen SNR conditions, we use a distribution fitting method. For unseen noise conditions, we use a VAD algorithm to produce noise-only frames and determine the thresholds from a small development set. Auditory segmentation is incorporated to further improve the rethresholded mask. The experiments and comparisons show that the proposed approach achieves good generalization in unmatched conditions.

CHAPTER 4

LEARNING INVARIANT FEATURES FOR SPEECH SEPARATION

In the previous chapter, a rethresholding approach is used to address the generalization problem for supervised IBM estimation. However, rethresholding can only be performed after seeing a certain length of new mixtures, and thus is not suitable for real-time applications. We propose to use a novel metric learning method to learn invariant speech features in the kernel space. As the learned features encode speech-related information that is robust to different noise types, the system is expected to generalize to unseen noise conditions. The work presented in this chapter has been published in the *Proceedings of the 2013 IEEE International Conference on Acoustic, Speech, and Signal Processing* [39].

4.1 Introduction

One issue in supervised classification is that the training data and the test data are expected to extracted from the same distribution. When the distribution changes, the trained models may not produce reasonable results in the test dataset. To generalize a speech separation system to unseen noise conditions, one can build a massive training set including a large variety of noises. However, such training is very computationally expensive and it would be impossible to include all noises in a training set. Previous chapter described a rethresholding approach for generalization, which is a model adaption approach and can be performed only after seeing certain length of new mixtures.

In this chapter, we use a more desirable approach to address the generalization problem. We propose to learn invariant speech features in the kernel space using Information-theoretic Metric Learning (ITML) [20]. Because the learned kernel encodes invariant information related only to speech, a classifier trained on this kernel should be able to generalize to unseen noise types. We train an SVM based on the learned kernels and successfully classify test data under new noise conditions. Note that we only consider speech separation from non-speech interference in this study.

In the next section, we relate our approach to existing work on speech separation and metric learning. The overall framework of the system is given in Section 4.3. Section 4.4 describes how to learn the kernel and incorporate it into the SVM. We evaluate the system in Section 4.5 and conclude in Section 4.6.

4.2 Related work

Supervised learning has been recently used to classify T-F units, including MLP [63], GMM [70], and SVM [37], [59]. These approaches mostly deal with the situations in which the test noises are included in the training set. However, if noises are not seen in the training phase, the probabilistic properties of the extracted features in the test set may differ significantly from those in the training set and the trained models may not work well under these noise conditions.

In machine learning, transfer learning and domain adaptation aim to compensate for data shift, i.e., a change in the feature distribution from the training set to the test set [102]. Relevant methods have been developed in the natural language processing (NLP) [19] and computer vision communities [114], [74], [23], which can be roughly categorized as classifier adaptation and feature transformation. The former approach utilizes the target domain information to adapt the parameters of classifiers [23]. In the speech separation field, Ozerove *et al.* [96] and our previous study [38], [40] utilize noise only intervals to collect noise information for model adaptation. Because the adaptation needs to detect the noise intervals in the test mixtures, it is difficult to apply to real-time processing.

On the other hand, feature transformation utilizes metric learning methods to transfer the input features between domains and then apply a classifier [19], [114], [74]. The advantage of this approach is that the learned features can be domain-independent, which enables it to deal with novel problems with new feature types or dimensionalities [104], [74]. For speech separation, one important property is that the features extracted from speech are usually much more stable than those from noises. In other words, if we can capture the common speech characteristics independent of noise types, it is possible to utilize them to separate speech under various noise conditions. In this chapter, we learn invariant speech features across different noise conditions, which allow for generalization to new noises without any prior knowledge of the noise.

4.3 Speech separation using kernel SVM

4.3.1 Feature extraction

An input signal $s(t)$ is first passed through a 64-channel gammatone filterbank spanning from 80 Hz to 5000 Hz. The response of each filter channel is then divided into 20-ms time frames with 10-ms frame shift, forming a cochleagram [134]. We use $u_{c,m}$ to denote a T-F unit for frequency channel c and time frame m. For each T-F unit, we extract acoustic features including AMS, RASTA-PLP, mel-frequency cepstral coefficients (MFCC), and pitch-based features. Further, for every dimension of the features, we calculate delta features across time frames and frequency channels to capture variation information. The concatenation of these features have been proven to be effective in speech separation [137] and are used in this chapter.

4.3.2 SVM classification with learned kernels

Because of the different spectral properties of speech, we train an SVM in each channel to estimate the IBM. Previous studies directly use extracted features to train the SVM and yield accurate classification results under matched noise conditions [37], [137].

In order to generalize the system to unseen noise conditions, we aim to learn a non-linear transformation $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ to map original features into a high dimensional space, where d and d' denote the dimensionality of the original space and the kernel space respectively. Here, the underlying idea of the feature transformation is that for two data points from different noise conditions (domains), the learned transformation should maximizes the distances between them if they have different labels and minimizes the distances if they have the same label. This class-based cross-domain constraint will be applied during the transformation learning.

Furthermore, because the SVM can be viewed as a kernel machine, instead of explicitly computing $\phi(\mathbf{x})$, we only need to compute a kernel function κ such that $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ [127]. Therefore, we first learn a kernel using data from multiple noise conditions and then apply the learned kernel to the SVM for supervised learning. In the test phase, each data point is also kernelized for classification. We will discuss kernel learning in detail in the next section.

Finally, the SVM labels T-F units in each channel to form an estimated IBM. The separated speech is resynthesized using the cochleagram of the mixture and the estimated IBM [134].

4.4 Domain-invariant kernel learning

4.4.1 Cross-domain constraints

In this section, we discuss how to learn domain-invariant features in the kernel space. For a general metric learning problem, given a data set $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \mathbf{x}_i \in \mathbb{R}^d$, one aims to learn an appropriate Mahalanobis distance parameterized by a positive definite matrix W between $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$d_W(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T W (\mathbf{x}_i - \mathbf{x}_j) \tag{4.1}$$

Since W is symmetric positive definite, by factorizing W as $W = G^T G$, we can equivalently view the distance $d_W = ||G\mathbf{x}_i - G\mathbf{x}_j||^2$, that is, the transformation G serves as a linear transformation applying to data points.

Since the linear transformation is not powerful enough for our application, we are interested in working in the kernel space, where we use a non-linear function ϕ to map input into a high-dimensional space. Then, the distance is:

$$d_W(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) = (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))^T W (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \tag{4.2}$$

To learn the desired metric, we use the data to create pairwise similarity and dissimilarity constraints. To improve the generalizability in our study, we generate the constraints across different domains based on the labels. Suppose that the training set consists of multiple domains $\mathcal{D}_m, m = 1, \ldots, M$, corresponding to multiple noise conditions, and a data point in the domain $\mathcal{D}_m$ is denoted as $\mathbf{x}_i^{\mathcal{D}_m}$ with its label $y_i^{\mathcal{D}_m}$. To learn the domain-invariant transformation, we use the following cross-domain constraints. For a pair of data points $\mathbf{x}_i$ and $\mathbf{x}_j$ from two different domains $\mathcal{D}_a$ and $\mathcal{D}_b$, we create the constraints:

$$\begin{aligned} &d_W(\phi(\mathbf{x}_i^{\mathcal{D}_a}), \phi(\mathbf{x}_j^{\mathcal{D}_b})) \leq u, \text{ if } y_i^{\mathcal{D}_a} = y_j^{\mathcal{D}_b} \\ &d_W(\phi(\mathbf{x}_i^{\mathcal{D}_a}), \phi(\mathbf{x}_j^{\mathcal{D}_b})) \geq l, \text{ if } y_i^{\mathcal{D}_a} \neq y_j^{\mathcal{D}_b} \end{aligned} \tag{4.3}$$

where u and l are parameters representing the distance thresholds. As we create cross-domain constraints for every pair of domains, there are totally $\binom{M}{2}$ pairs of domains for constraints.

These cross-domain constraints enforce the algorithm to learn a metric such that the data points with the same label should be close to each other no matter which domains they belong to. By applying the constraints to every pair of domains, the learned transformation captures not only the domain shift between any two of them but also the common information shared by all these domains. Since the data in different domains correspond to speech mixed with different noises, the transformation presumably encodes speech-related information that is independent to noise types.

4.4.2 Kernel learning with ITML

Given the constraints in Eq. (4.3), our problem is to learn a positive-definite matrix W that parameterizes the Mahalanobis distance. We adopt the ITML [20] algorithm and discuss its kernelized version in this subsection. The algorithm uses the LogDet divergence D_{ld} to regularize W against a specified positive definite matrices W_0:

$$D_{ld}(W, W_0) = \text{ trace}(WW_0^{-1}) - \log\det(WW_0^{-1}) \tag{4.4}$$

and the metric learning problem is:

$$\begin{aligned}
&\min_{W\succeq 0} D_{ld}(W, W_0) \\
&\text{s.t.}\ \ d_W(\phi(\mathbf{x}_i^{\mathcal{D}_a}), \phi(\mathbf{x}_j^{\mathcal{D}_b})) \le u,\ \text{if}\ y_i^{\mathcal{D}_a} = y_j^{\mathcal{D}_b} \\
&\qquad d_W(\phi(\mathbf{x}_i^{\mathcal{D}_a}), \phi(\mathbf{x}_j^{\mathcal{D}_b})) \ge l,\ \text{if}\ y_i^{\mathcal{D}_a} \ne y_j^{\mathcal{D}_b} \\
&\qquad a, b \in \{1, \dots, M\}
\end{aligned} \tag{4.5}$$

Therefore, we are interested in finding a metric W that is close to an original metric W_0 but satisfies our desired constraints. Note that, we create the constraints for every pair of domains, which is different from previous cross-domain metric learning [74, 114], where only one pair of domains is considered.

We now consider kernelizing the problem. Given a set of data points, let K_0 denote the input kernel matrix for the data, that is, $K_0(i,j) = \kappa_0(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. In this study, we choose the Gaussian kernel to introduce nonlinearity, i.e., $\kappa_0(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})$. We use $K(i,j)$ to denote the kernel we want to learn, i.e., $K(i,j) = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T W \phi(\mathbf{x}_j)$. Therefore, according to Eq. (4.2), we have:

$$\begin{aligned}
&d_W(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) \\
&\ \ =\phi(\mathbf{x}_i)^T W \phi(\mathbf{x}_i) - 2\phi(\mathbf{x}_i)^T W \phi(\mathbf{x}_j) + \phi(\mathbf{x}_j)^T W \phi(\mathbf{x}_j) \\
&\ \ =K(i,i) + K(j,j) - 2K(i,j)
\end{aligned} \tag{4.6}$$

In addition, to avoid an infeasible solution in Eq. (4.5), we incorporate a slack variable $\boldsymbol{\xi}$ to provide a tradeoff between minimizing the divergence between K and

K_0 and satisfying the constraints. Finally, the non-linear metric learning problem can be formulated to a kernel learning problem:

$$\begin{aligned}
&\min_{K\succeq 0,\boldsymbol{\xi}} D_{ld}(K,K_0)+\gamma D_{ld}(\mathrm{diag}(\boldsymbol{\xi}),\mathrm{diag}(\boldsymbol{\xi}_0))\\
&\text{s.t. } K(i,i)+K(j,j)-2K(i,j)\le \xi_{i,j},\ \text{if } y_i=y_j\\
&\quad K(i,i)+K(j,j)-2K(i,j)\ge \xi_{i,j},\ \text{if } y_i\neq y_j\\
&\quad (\mathbf{x}_i,y_i)\in\mathcal{D}_a,(\mathbf{x}_j,y_j)\in\mathcal{D}_b,\ \text{and } a,b\in\{1,\ldots,M\}
\end{aligned} \tag{4.7}$$

where, γ is the tuning parameter. The entries in $\boldsymbol{\xi}_0$ are set to u for similarity constraints and l for dissimilarity constraints.

To solve this optimization problem, we follow the approach given in [20] which employs Bregman projections to iteratively compute the kernel [74]:

$$K_{t+1}\leftarrow K_t+\beta K_t(\mathbf{e}_i-\mathbf{e}_j)(\mathbf{e}_i-\mathbf{e}_j)^T K_t \tag{4.8}$$

where $\mathbf{e}_i$ is the standard basis vector with a 1 in the ith coordinate and β is a parameter computed in the algorithm.

Once we learn the kernel K, it is straightforward to use Eq. (4.6) to compute the distance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ that are in the training set. But for new data points $\mathbf{z}_1$ and $\mathbf{z}_2$ that are not in the training set, we need to compute the kernel function $\kappa(\mathbf{z}_1,\mathbf{z}_2)$. Here, we directly give the equation to compute the kernel for a pair of arbitrary data points $\mathbf{z}_1$ and $\mathbf{z}_2$:

$$\kappa(\mathbf{z}_1,\mathbf{z}_2)=\kappa_0(\mathbf{z}_1,\mathbf{z}_2)+\mathbf{k}_1^T K_0^{-1}(K-K_0)K_0^{-1}\mathbf{k}_2 \tag{4.9}$$

Here, $\mathbf{k}_i=[\kappa_0(\mathbf{z}_i,\mathbf{x}_1),\ldots,\kappa_0(\mathbf{z}_i,\mathbf{x}_n)]^T$, and $\mathbf{x}_i$ is the data point in the training set used to learn the kernel. For details of the kernel learning algorithm, see [20] and [61].

4.5 Experiments

We now evaluate our kernel learning based separation system. The IEEE corpus [110] is used to train and test the system. The input SNR is -5 dB and LC is set to -10 dB, which pose a very challenge problem. To learn the domain-invariant kernel, we first choose 10 utterances mixed with 5 types of noise out of a 100 non-speech noise corpus [52]. Thus, there are around 3,000 data points for each noise condition. We randomly choose a subset of around 100 data points in each condition to create the cross-domain constraints, so $100 \times 100 \times \binom{5}{2} = 100,000$ pairs of constraints are used in the kernel learning. We set the distance thresholds u and l to 5% and 95% percentile of the distribution of the observed distances between pairs of points respectively. The slack variable γ and the variance of the Gaussian kernel σ are tuned using cross validation. After we learn the kernel, we train the SVM using another 30 utterances mixed with the same 5 noises. According to Eq. (4.9), we compute the kernel for these data for SVM training.

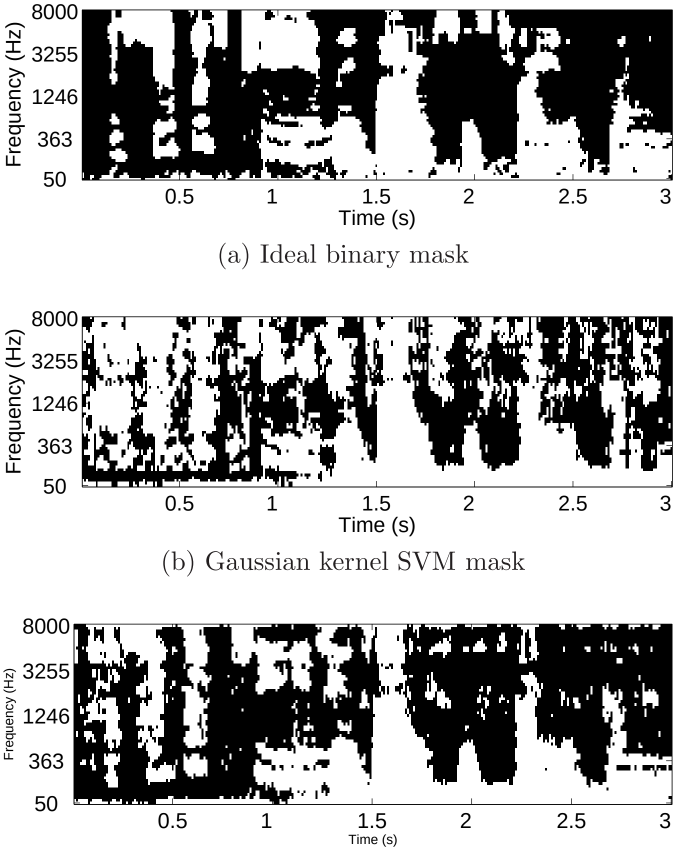
To test the system, we use 10 utterances mixed with 12 types of noise—N1: white noise, N2: cocktail party noise, N3: rock music, N4: telephone, N5: fan noise, N6: clock alarm, N7: traffic noise, N8: crowd noise with clap, N9: bird chirping with water flowing, N10: wind noise, N11: rain noise, N12: babble noise. The test noises cover both stationary and non-stationary noises and have very different frequency characteristics. None of the utterances and the noises are seen in the kernel learning and SVM training phase.

As an example, Fig. 4.1 illustrates mask estimation results for an utterance mixed with an unseen crowd noise with clap at -5 dB using the SVM with the Gaussian kernel and the SVM with the learned domain-invariant kernel respectively. It is clear that the Gaussian kernel SVM leads to severe classification errors because the noise is significantly different from those in the training set. By using kernel learning the system yields a substantially better mask due to the robustness of the learned kernel against different noise types.

We compute HIT−FA to quantify the performance of our system. Table 4.1 shows the average classification accuracy and the HIT−FA rates over all 12 noises. We compare the Gaussian kernel SVM (G-SVM) and the SVM with learned domain-invariant kernel (KL-SVM). In the left two columns of the table, in order to eliminate the impact of pitch errors we use ground-truth pitch extracted from the premixed speech [10] to generate the pitch-based features. In the right two columns, we use a pitch estimator [32] to extract pitch from mixtures. Both experiments clearly show that learning the domain-invariant kernel significantly boosts the classification accuracy and the HIT−FA rates under new noise conditions.

Table 4.1: Average classification accuracy and HIT−FA rates.

	Ground-truth Pitch		Estimated Pitch	
	G-SVM	KL-SVM	G-SVM	KL-SVM
Accuracy	0.742	0.794	0.703	0.746
HIT−FA	0.469	0.537	0.390	0.456



(a) Ideal binary mask

(b) Gaussian kernel SVM mask

(c) Domain-invariant kernel SVM mask

Figure 4.1: IBM estimation results. (a) IBM for the mixture. (b) Estimated IBM using the Gaussian kernel SVM. (c) Estimated IBM using the domain-invariant kernel SVM. White regions represent 1s and black 0s.

We further compare the proposed method with two other speech separation approaches. The first one is a state-of-the-art speech enhancement algorithm based on a minimum mean-squared error (MMSE) estimator proposed by Hendriks *et al.* [45]. The second one is our previous approach which uses the rethresholding technique to adapt the SVM classification under different noise conditions [40]. The proposed approach in this comparison uses the estimated pitch. As shown in Fig. 4.2, the proposed approach achieves the highest HIT−FA under every noise condition. On average, the proposed approach outperforms Hendriks *et al.* by 14 percentage points and our previous system by 4 percentage points. We point out that, our previous system needs noise information extracted from the test mixture to adapt the trained model, while the proposed approach can be directly applied to the test mixture and does not need to collect information from the new noise, which is a considerable advantage.

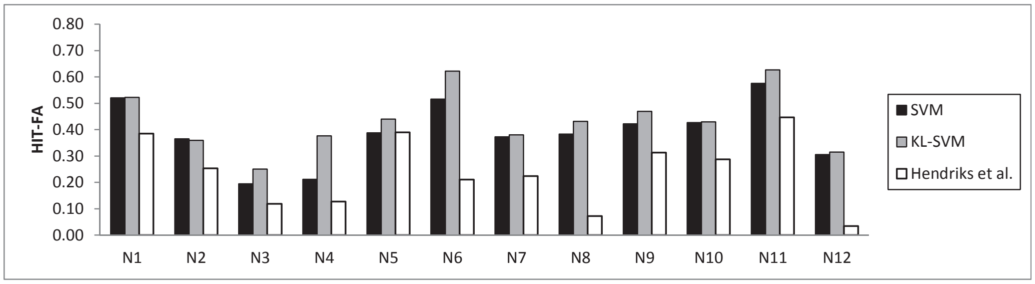


Figure 4.2: HIT−FA comparison under unseen noise conditions

4.6 Conclusion

In this study, we have proposed to learn a domain-invariant kernel to encode speech-related information that is robust to different noise types. With the learned kernel, the speech separation system can be applied to new noise conditions without any prior information of the noise.

CHAPTER 5

DEEP NEURAL NETWORKS BASED SPECTRAL MAPPING FOR SPEECH DEREVERBERATION AND DENOISING

The previous chapters mainly deal with speech separation from background noise. But, in real-world environments, human speech is usually distorted by both reverberation and background noise, which have negative effects on speech intelligibility and speech quality. They also cause performance degradation in many speech technology applications, such as automatic speech recognition. Therefore, the dereverberation and denoising problems must be dealt with in daily listening environments. In this chapter, we perform dereverberation and denoising using supervised learning. A DNN is trained to directly learn a spectral mapping from the magnitude spectrogram of corrupted speech to that of clean speech. The proposed approach substantially attenuates the distortion caused by reverberation and noise, which is simple but yet effective. Systematic experiments show that the proposed approach leads to significant improvement of predicted speech intelligibility and quality scores, as well as speech recognition rates in reverberant noisy conditions. Part of the material presented in this chapter has been published in the *Proceedings of the 2014 IEEE International Conference on Acoustic, Speech, and Signal Processing* [41].

5.1 Introduction

In real-world environments, the sound reaching the ears comprises the original source (direct sound) and its reflections from various surfaces. These attenuated, time-delayed reflections of the original sound combine to form a reverberant signal. In reverberant environments, speech intelligibility is degraded substantially for hearing impaired listeners [73], and normal hearing listeners when reverberation is severe [109]. In addition, room reverberation when combined with background noise is particularly disruptive for speech perception. Reverberation and noise also cause significant performance degradation in ASR [72] and SID systems [113], [150]. Given the prevalence of reverberation and noise, a solution to the dereverberation and denoising problems will benefit many speech technology applications.

Reverberation corresponds to a convolution of the direct sound and the room impulse response (RIR), which distorts the spectrum of speech in both time and frequency domains. Thus, dereverberation may be treated as inverse filtering. The magnitude relationship between an anechoic signal and its reverberant version is relatively consistent in different reverberant conditions, especially within the same room. Even when reverberant speech is mixed with background noise, it is still possible to restore speech to some degree from the mixture, because speech is highly structured. These properties motivate us to utilize supervised learning to model the reverberation and mixing process. In this chapter, we propose to learn the mapping from the corrupted speech to its anechoic, premixed version. The mapper can be trained where the input is the spectral representation of the corrupted speech and the desired output is that of the anechoic clean speech.

DNNs have shown strong learning capacity [49]. A stacked denoising autoencoder (SDA) [130] is a deep learning method, and it can be trained to reconstruct the raw clean data from the noisy data, where hidden layer activations are used as learned features. Although SDAs were proposed to improve generalization, the main idea behind SDAs motivated us to utilize DNNs to learn the mapping from the corrupted data to clean data. A recent study [139] used DNNs to denoise acoustic features in each time-frequency unit for speech separation. Our approach, on the other hand, deals with reverberant and noisy speech and the mapping directly applies to frame-level spectral features.

The chapter is organized as follows. In the next section, we discuss related speech dereverberation and denoising studies. We then describe our approach in detail in Section 5.3. The experimental results are shown in Section 5.4. We discuss related issues and conclude the chapter in the last section.

5.2 Relation to prior work

Many previous approaches have been proposed to deal with speech dereverberation [93]. Inverse filtering is one of the commonly used techniques [87]. Since the reverberation effect can be described as a convolution of clean speech with the room impulse response, the inverse filtering based approach first determines an inverse filter that can reverse the effects of the room response, and then estimates the anechoic signal by convolving the reverberant signal with the inverse filter. However, in many situations, the inverse filter cannot be determined directly and must be estimated, which is a hard problem. Further, this approach assumes that the RIR function is minimum-phase that is often not satisfied in practice [94]. Wu and Wang [146] utilized a two-stage approach including inverse filtering and spectral subtraction to deal with early reverberation and late reverberation separately, which relies on an accurate estimate of the inverse filter in one microphone scenarios. Other studies dealt with dereverberation by exploiting the properties of speech such as modulation spectrum [5], and harmonic structure [145], [106].

Recent studies show that the IBM can be extended to suppress reverberation and improve speech intelligibility [73], [108], [109]. The IBM based approaches treat the direct sound or direct sound plus the early reflections as the target and the rest as the masker, and the dereverberated signal is resynthesized from the binary mask. Therefore, the IBM can still be considered as an effective computational goal for dereverberation. Hazrati *et al.* [44] proposed to estimate a binary mask based on a single variance-based feature against an adaptive threshold and yielded intelligibility improvements for cochlear implantees. In principle, the IBM based approach can deal with both reverberation and noise simultaneously; however, few previous studies aim to estimate the IBM for both dereverberation and denoising. Jin and Wang [63] use an MLP to estimate the IBM for speech separation but the target is the reverberant noise-free speech.

5.3 Algorithm description

We describe the algorithm in this section, including three subsections: feature extraction, model training, and post-processing.

5.3.1 Spectral features

We first extract features for spectral mapping. Given a time domain input signal $s(t)$, we use short time Fourier transform (STFT) to extract features. We first divide the input signal into 20-ms time frames with 10-ms frame shift, and then apply fast Fourier transform (FFT) to compute log spectral magnitudes in each time frame. For a 16 kHz signal, we use 320-point FFT and therefore the number of frequency bins is 161. We denote the log magnitude in the kth frequency and the mth frame as $X(m, k)$. In order to incorporate temporal dynamics, we include the spectral features of neighboring frames into a feature vector. Therefore, the input feature vector for the DNN feature mapping is:

$$\tilde{\mathbf{x}}(m) = [\mathbf{x}(m-d), \ldots, \mathbf{x}(m), \ldots, \mathbf{x}(m+d)]^T \tag{5.1}$$

where d denotes the number of neighboring frames in each side and is set to 5 in this study. So the dimensionality of the input is $161 \times 11 = 1771$.

The desired output of the neural network is the spectrogram of clean speech in the current frame m, denoted by a 161-dimensional feature vector $\mathbf{y}(m)$, whose elements correspond to the log magnitude in each frequency bin at the mth frame.

5.3.2 DNN based spectral mapping

We train a deep neural network to learn the spectral mapping from reverberant, or reverberant and noisy, signals to clean signals.

The DNN in this study includes three hidden layers, as shown in Fig. 5.1. The input for each training sample is the log magnitude spectrogram in a window of frames, and the number of input units is the same as the dimensionality of the feature vector. The output is the log magnitude spectrogram in the current frame, corresponding to 161 output units. Each hidden layer includes 1600 hidden units. The number of hidden layers and hidden units are chosen from a development set.

The objective function for optimization is based on mean square error. Eq. 5.2 is the cost for each training sample:

$$\mathcal{L}(\mathbf{y}, \mathbf{x}; \boldsymbol{\Theta}) = \sum_{c=1}^{C} (y_c - f_c(\mathbf{x}))^2 \tag{5.2}$$

where $C = 161$ corresponds to the index of the highest frequency bin, $\mathbf{y} = (y_1, \ldots, y_C)^T$ is the desired output vector, and $f_c(\cdot)$ is the actual output of the cth neuron in the output layer. $\boldsymbol{\Theta}$ denotes the parameters we need to learn. To train the neural network, the input is normalized to zero mean and unity variance, and the output is normalized into the range of $[0, 1]$. The activation function in the hidden layers is

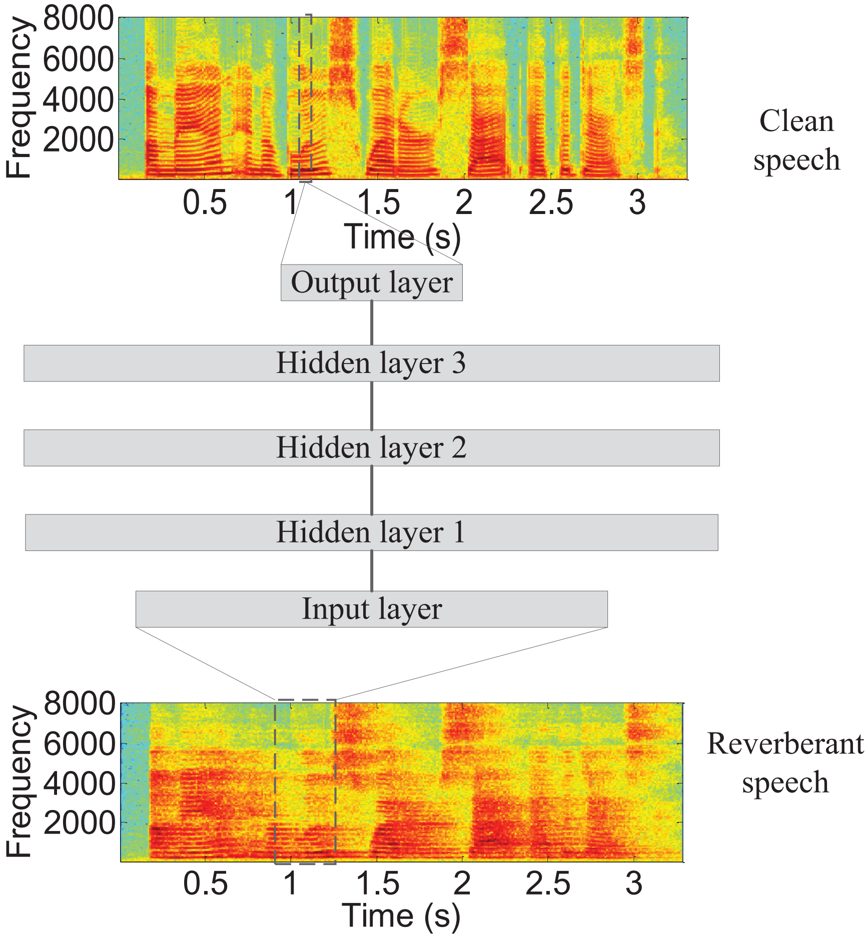


Figure 5.1: Structure of the DNN based spectral mapping.

the rectified linear function and the output layer uses the sigmoid function, shown in Eqs. 5.3 and 5.4 respectively:

$$f(x) = \max(0, x) \tag{5.3}$$

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5.4}$$

The weights of the DNN are randomly initialized without pretraining. We use backpropagation with mini-batch stochastic gradient descent to train the DNN model, and the actual cost in each mini-batch is computed from the summation over multiple training samples using Eq. 5.2. The optimization technique uses adaptive gradient descent along with a momentum term [24].

The output of DNN is the estimated log magnitude spectrogram of clean speech. With the capacity of learning internal representations, DNN promises to be able to encode the spectral transformation from corrupted speech to clean speech and help to restore the magnitude spectrogram of clean speech.

5.3.3 Post-processing

After DNN generates magnitude spectrogram estimates, we need to resynthesize time-domain signals using inverse FFT.

A straightforward method to reconstruct time-domain signals is to directly apply inverse short-time Fourier transform (iSTFT) using the DNN-generated magnitude and the phase from unprocessed time-domain signals. However, the original phase of noise-free speech is corrupted, and the corruption usually introduces perceptual disturbances and leads to negative effects on sound quality. In addition, STFT is computed by concatenating Fourier transforms of overlapping frames of a signal, and thus is a redundant representation of the time-domain signal. For a spectrogram-like matrix in the time-frequency domain, it is not guaranteed there exists a time-domain signal whose STFT is equal to that matrix [34], [76]. In other words, the magnitude spectrogram of the resynthesized time-domain signal could be different from the one we intended to resynthesize a signal from. This inconsistency should be taken into account for synthetic or modified spectrograms, like our DNN-generated magnitudes.

In order to minimize the incoherence between the phase and the magnitude from which we want to reconstruct a signal, we use an iterative procedure to reconstruct a time-domain signal as given in Algorithm. 1 [34]:

Algorithm 1 Iterative signal reconstruction

```
Input: Target magnitude $Y^0$, noisy phase $\phi^0$ and iteration number $N$
Output: Time-domain signal $s$
$Y \leftarrow Y^0$, $\phi \leftarrow \phi^0$, $n \leftarrow 1$
while $n \leq N$ do
    $s^n \leftarrow \text{iSTFT}(Y, \phi)$
    $(Y^n, \phi^n) \leftarrow \text{STFT}(s^n)$
    $Y \leftarrow Y^0$
    $\phi \leftarrow \phi^n k$
    $n \leftarrow n + 1$
end while
$s \leftarrow s^N$
```

Here, $N = 20$ in our study. The algorithm iteratively updates the phase ϕ at each step by replacing it with the phase of the STFT of its inverse STFT, while the target magnitude Y^0 is the DNN-generated output, which is always fixed. The iteration aims to find the closest realizable magnitude spectrogram consistent with the given magnitude spectrogram.

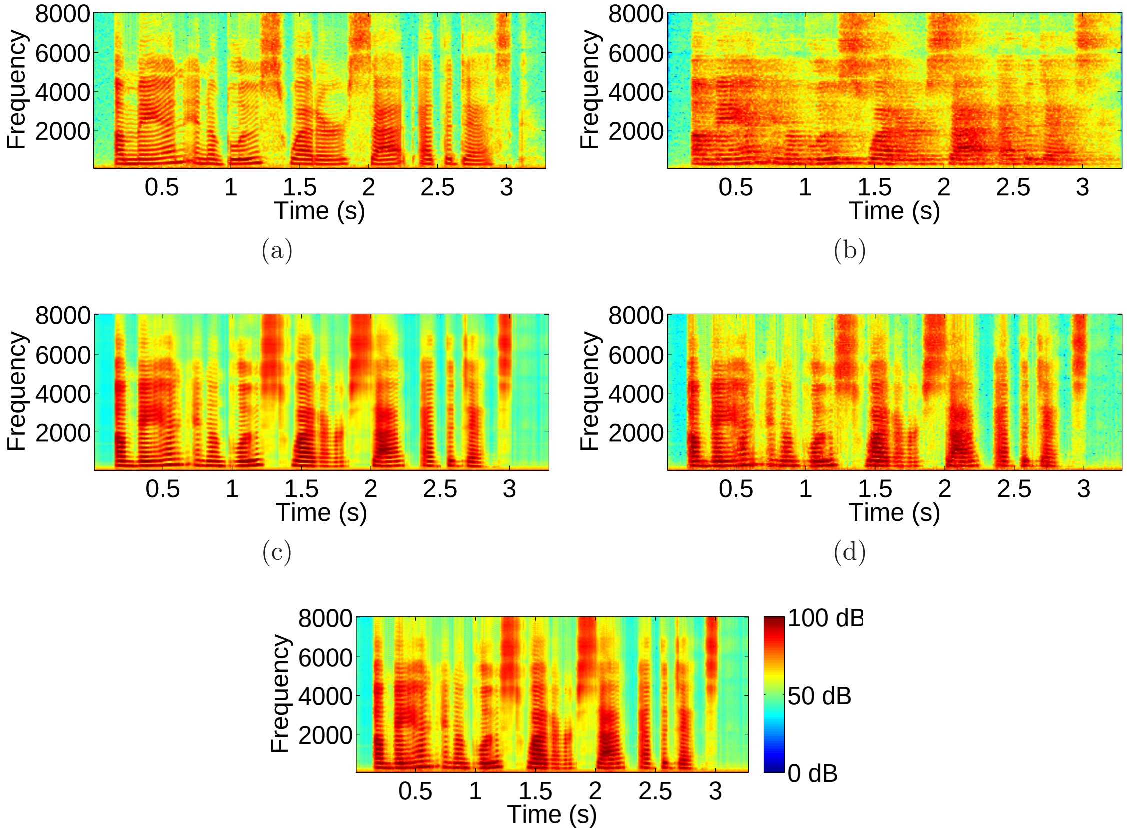
We use above post-processing to reconstruct a time-domain signal as a waveform output of our system.

Fig. 5.2 shows an example of the spectral mapping for a female sentence "A man in a blue sweater sat at the desk". Figs. 5.2(a) and (b) show the log magnitude spectrogram of the clean speech and the reverberant speech with $T_{60} = 0.6$ s. The corresponding DNN output is shown in Fig. 5.2(c). As shown in Fig. 5.2(c), the smearing energy caused by reverberation is largely removed or attenuated, and the boundaries between voiced and unvoiced frames are considerably restored, showing that the DNN output is a very good estimate of the spectrogram of the clean speech. Fig. 5.2(d) is the magnitude spectrogram of the time-domain signal resynthesized from the magnitude in Fig. 5.2(c) and reverberant phase. Comparing Figs. 5.2(c) with (d), the spectrogram in Fig. 5.2(d) is not as clean as the DNN output in Fig. 5.2(c) because of the use of reverberant phase and inconsistency of STFT. Fig. 5.2(e) is the spectrogram of the time-domain signal using post-processing, where the spectrogram is improved by iterative signal reconstruction.

5.4 Experiments

5.4.1 Metrics and parameters

We quantitatively evaluate our approach by two objective measurements of speech intelligibility: frequency-weighted segmental speech-to-noise ratio (SNR$_{fw}$) [84] and short-time objective intelligibility measure (STOI) [122]. Specifically, SNR$_{fw}$ is a



(a)

(b)

(c)

(d)

(e)

Figure 5.2: DNN dereverberation results. (a) Log magnitude spectrogram of clean speech. (b) Log magnitude spectrogram of reverberant speech with $T_{60} = 0.6$ s. (c) DNN outputs. (d) Log magnitude spectrogram of resynthesized signal. (e) Log magnitude spectrogram of resynthesized signal with post-processing.

speech intelligibility indicator, computing a signal-to-noise estimate for each critical band:

$$\mathrm{SNR}_{fw} = \frac{10}{M} \sum_{m=1}^{M} \frac{\sum_{k=1}^{K} W(k) \log_{10} \frac{|S(m,k)|^2}{|S(m,k) - \hat{S}(m,k)|}}{\sum_{k=1}^{K} W(k)} \tag{5.5}$$

where $W(k)$ is the weight placed on the kth frequency band, K is the number of bands, M is the total number of frames in the signal, $S(m,k)$ is the critical-band magnitude of the clean signal in the kth frequency band at the mth frame, and $\hat{S}(m,k)$ is the corresponding spectral magnitude of the processed signal in the same band.

STOI is recently proposed to predict speech intelligibility. It computes the correlation between temporal envelopes of the clean and processed speech in short-time segments as an intelligibility indicator, ranging from 0 to 1. STOI has been shown to have high correlation with speech intelligibility of human listeners [122].

In addition, we evaluate speech quality using Perceptual Evaluation of Speech Quality (PESQ) [105], which computes disturbance between clean speech and processed speech using cognitive modeling as a speech quality score. The range of PESQ score is from -0.5 to 4.5.

As we mentioned in Section 5.3.1, we utilize context information using a concatenation of features from 5 frames in each side of the current frame. Temporal information is an important property for speech signals, and thus adding these neighboring frames should be helpful to learn a spectral mapping. We have conducted experiments using different window sizes. Comparing with the 11-frame window, the SNR_{fw} results of a 7-frame window and a 3-frame window degrade by around 0.5 dB and 2.5 dB, respectively.

The architecture of the DNN influences its learning performance. We have conducted experiments using different numbers of hidden layers. A DNN with three hidden layers performs slightly better than that with two hidden layers in terms of SNR_{fw} (by 0.2 dB), and better than that with a single hidden layer (by 1.1 dB).

5.4.2 Dereverberation

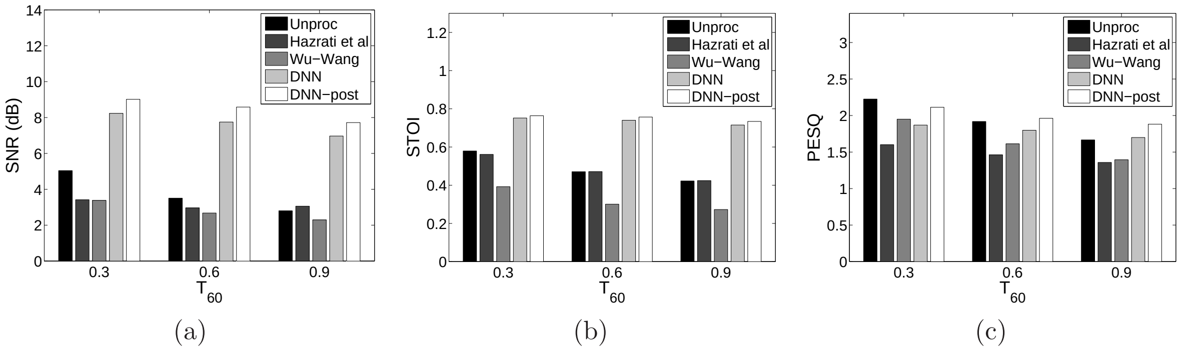


Figure 5.3: DNN based dereverberation results: (a) SNR$_{fw}$, (b) STOI, (c) PESQ. “Unproc” denotes the results for unprocessed reverberant speech. ‘Hazrati et al.” and “Wu-Wang” denote two baselines as described. “DNN” denotes the proposed spectral mapping approach without post-processing. “DNN-post” denotes the proposed spectral mapping approach with iterative signal reconstruction processing.

We first evaluate dereverberation performance in this section. To mimic room acoustics, we generate a simulated room corresponding to a specific T_{60} [35] and randomly create a set of RIRs under this T_{60} condition. To train the system, we use three reverberation times of 0.3, 0.6, and 0.9 s, and for each T_{60} we generate 2 different RIRs. We use 200 anechoic utterances from the IEEE corpus [110] to form the training set. Therefore, there are $200 \times 3 \times 2 = 1200$ reverberant sentences in the training set. The test set includes 60 reverberant sentences, corresponding to 20 speech utterances, three T_{60}s, and one RIR. Neither the utterances nor RIRs are used in the training set.

We compare the proposed approach with two dereverberation algorithms. Hazrati *et al.* [44] proposed a recent dereverberation approach, utilizing a variance-based feature from the reverberant signal and comparing its value against an adaptive threshold to compute a binary mask for dereverberation. Wu and Wang [146] used estimated inverse filters and spectral subtraction to attenuate early reverberation and late reverberation, respectively.

In Fig. 5.3, we show the evaluation results in terms of frequency-weighted SNR, STOI, and PESQ, as well as those of the comparison systems. For SNR_{fw} results shown in Fig. 5.3(a), the DNN based approach significantly improves SNR_{fw} relative to the unprocessed reverberant speech by 4 dB on average. The post-processing further boosts SNR_{fw} by around 1 dB. Comparing with Hazrati *et al.* and Wu and Wang, our DNN based methods achieve highest SNR_{fw} scores. Consistent with SNR_{fw}, Fig. 5.3(b) shows that the proposed methods yield high STOI scores under each reverberation time, higher than the unprocessed and the other two approaches by more than 0.25. As shown in Fig. 5.3(c), the proposed approach does not boost PESQ scores for the conditions of $T_{60} \leq 0.6$ s, partly because mild reverberation does not lead to significant sound quality degradation. When the reverberation time is long, the PESQ score is boosted by our approach as shown in the condition of $T_{60} = 0.9$ s.

Since our approach is a supervised learning method, it is important to evaluate its generalizability. We generate another set of RIRs with T_{60} from 0.2 to 1.0 s, with the increment of 0.1 s. Note that, none of RIRs in this experiment are seen in

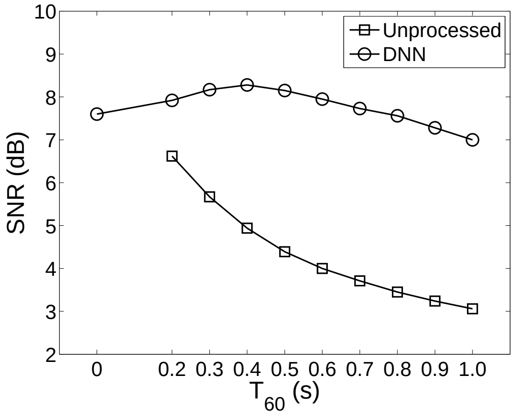
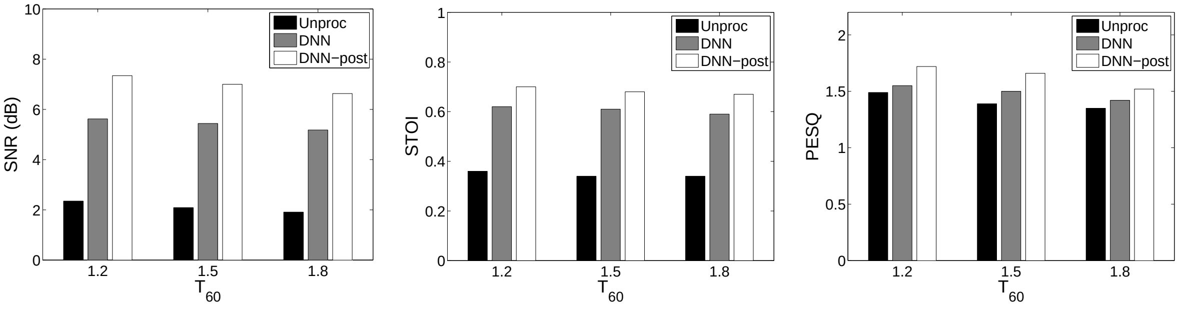


Figure 5.4: Generalization results in different T_{60} s. "DNN" denotes the DNN based spectral mapping approach without post-processing, and "Unprocessed" the results for original reverberant speech.

the training set as they are created from different rooms. We compare unprocessed signals with our DNN based approach without post-processing. Fig. 5.4 shows the generalization results of SNR_{fw} for different T_{60}s. Compared with the unprocessed reverberant speech, the proposed approach substantially improves SNR_{fw} in each T_{60} and the advantage becomes increasingly larger as T_{60} increases, demonstrating that our approach generalizes well to new reverberant environments in a wide range. Fig 5.4 also shows the DNN processed results for anechoic speech, corresponding to $T_{60} = 0$ s in the figure.

Although mild to moderate reverberation does not significantly impact speech perception for normal hearing listeners, an adverse effect occurs when reverberation is severe [109]. We have also conducted dereverberation experiments for strong reverberation conditions, when T_{60} s is greater than 1.0 s. Similar to the above experiment,



(a) (b) (c)

Figure 5.5: DNN based dereverberation results under strong reverberation conditions: (a) SNR$_{fw}$, (b) STOI, (c) PESQ. "Unproc" denotes the results for unprocessed reverberant speech. "DNN" denotes the proposed spectral mapping approach without post-processing. "DNN-post" denotes the proposed spectral mapping approach with iterative signal reconstruction processing.

we use the same utterances to generate reverberant sentences with T_{60} set to 1.2 s, 1.5 s, and 1.8 s. The training and test sets use different utterances and different RIRs. Experimental results are shown in Fig. 5.5. Comparing with unprocessed sentences, the DNN based methods significantly improve SNR$_{fw}$ and STOI scores. Note that, unlike moderate reverberation conditions as shown in Fig. 5.3, PESQ scores are boosted in each reverberation time as shown in Fig. 5.5(c). In these conditions, the post-processing achieves consistently better performance for each metric.

5.4.3 Dereverberation and denoising

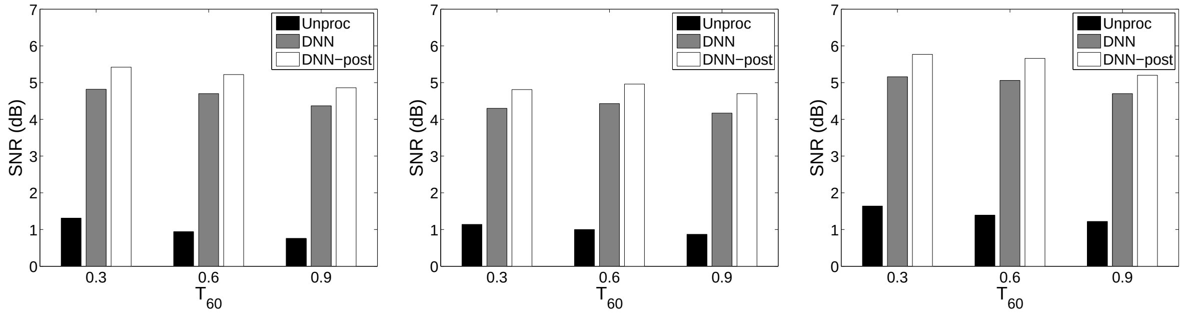
Our approach can deal with not only reverberation but also background noise. We can use the same supervised approach to perform dereverberation and denoising simultaneously. In this situation, the input to the neural network is the log magnitude spectrogram of reverberant and noisy speech, and the output is the log magnitude spectrogram of anechoic clean speech.

We conduct experiments for dereverberation and denoising. We generate a simulated room corresponding to a specific T_{60} and randomly create a set, $\{r_T, r_I, r_M\}$, representing the locations of the target, the interference and the microphone inside the room, respectively [63]. From these locations, a reverberant mixture $r(t)$ is constructed by

$$r(t) = h_T(t) * s(t) + \alpha h_I(t) * n(t) \tag{5.6}$$

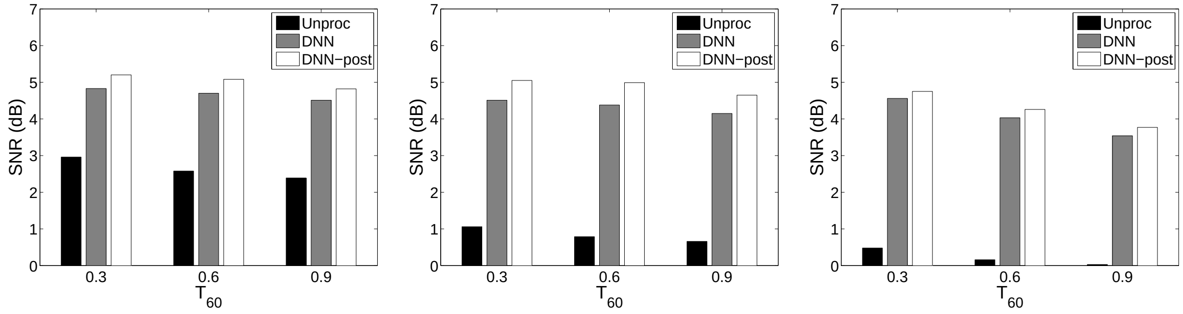
where, $h_T(t)$ and $h_I(t)$ are the RIR of the target and the interference at the microphone location, respectively. "$*$" denotes convolution. We use α as a coefficient to control the SNR of the mixture.

We simulate three acoustic rooms with different sizes and their T_{60}s are 0.3, 0.6, and 0.9 s, respectively. The training set contains reverberant mixtures including 600 reverberant sentences mixed with 3 noise types: speech-shaped noise, factory noise and babble noise [70] at 0 dB SNR. Here, the SNR is computed as the ratio of the energy of reverberant noise-free signal to that of reverberant noise-only signal. To test the system, 60 new reverberant utterances are mixed with the three training noises and three new noises, white noise, cocktail party noise, and crowd noise in playground [53], under each T_{60} but using different RIRs.



(a) (b) (c)

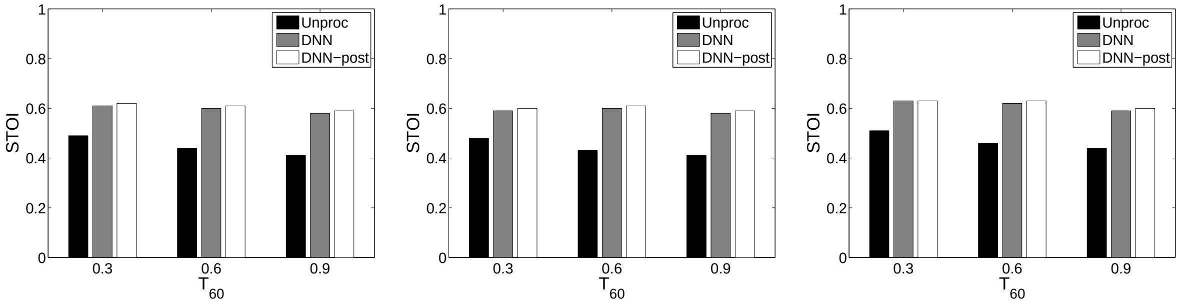
Figure 5.6: SNR$_{fw}$s for seen noises: (a) babble noise, (b) factory noise, (c) speech-shaped noise. "Unproc" denotes the results for unprocessed reverberant speech. "DNN" denotes the proposed spectral mapping approach without post-processing. "DNN-post" denotes the proposed spectral mapping approach with iterative signal reconstruction processing.



(a) (b) (c)

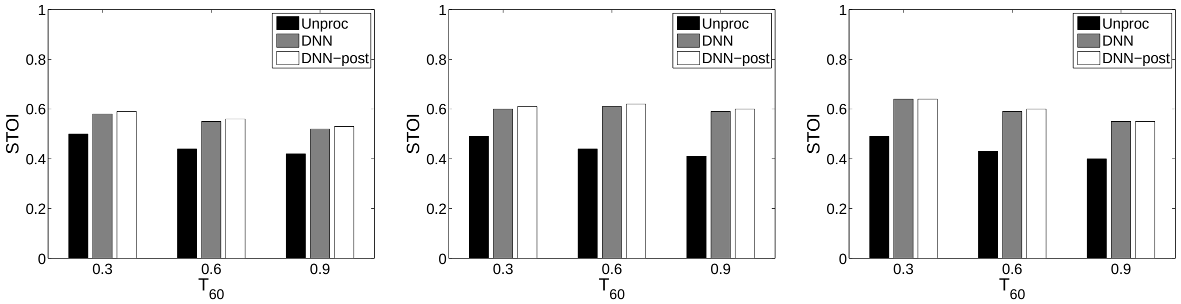
Figure 5.7: SNR$_{fw}$s for new noises:: (a) white noise, (b) cocktail-party noise, (c) crowd noise in playground.

Fig. 5.6 and Fig. 5.7 show SNR$_{fw}$ results for seen noises and new noises, respectively. The DNN based method increases SNR$_{fw}$ by 4.5 dB for seen noises and post-processing further yields 0.5 dB improvement. The proposed methods also achieve significant improvement for new noises, and the average advantage is around 3 dB, showing good generalization of the proposed approach.



(a) (b) (c)

Figure 5.8: STOI scores for seen noises: (a) babble noise, (b) factory noise, (c) speech-shaped noise.

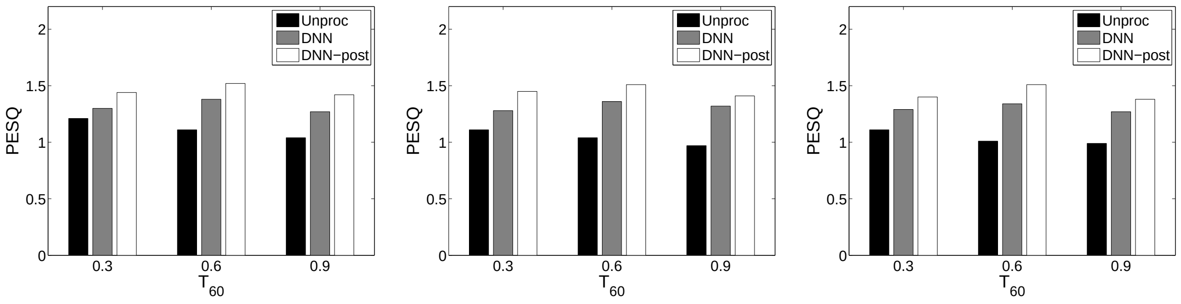


(a) (b) (c)

Figure 5.9: STOI scores for new noises:: (a) white noise, (b) cocktail-party noise, (c) crowd noise in playground.

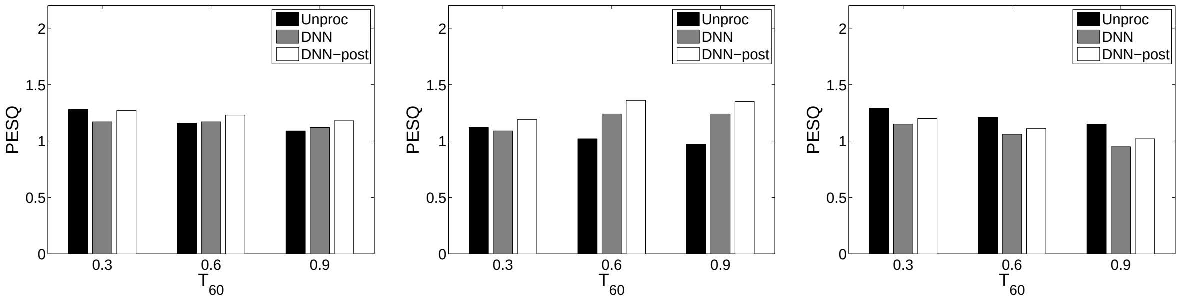
STOI scores are shown in Fig. 5.8 and Fig. 5.9, and DNN and DNN with post-processing have similar performances. On average, both increase STOI scores by around 0.15 for seen noises and 0.13 for new noises.

As shown in Fig. 5.10 and Fig. 5.11, PESQ results are improved by the proposed approach for both seen noises and new noises. For seen noises, the average PESQ scores for unprocessed, DNN, and DNN with post-processing sentences are 1.06, 1.31,



(a) (b) (c)

Figure 5.10: PESQ scores for seen noises: (a) babble noise, (b) factory noise, (c) speech-shaped noise.



(a) (b) (c)

Figure 5.11: PESQ scores for unseen noises:: (a) white noise, (b) cocktail-party noise, (c) crowd noise in playground.

1.45, respectively. For unseen noises, they are 1.13, 1.14, 1.21, respectively. These results demonstrate that the proposed approach improves speech quality when speech is corrupted by both noise and reverberation.

5.4.4 Robust speech recognition

The above evaluations show that our DNN based spectral mapping significantly attenuates reverberation and noise and produces good estimates of magnitude spectrogram of clean speech. As ASR algorithms only utilize magnitude spectrogram, our approach is expected to improve ASR performance in reverberant and noisy conditions.

In this evaluations, we use the 2nd CHiME challenge corpus to evaluate ASR performance [129]. In the CHiME-2 corpus, the utterances are taken from the speaker-independent 5k vocabulary subset of the Wall Street Journal (WSJ0) corpus. Each utterance is convolved with a fixed binaural room impulse response corresponding to a front position at a distance of 2 m, and then mixed with binaural recordings of real room noise over a period of days in the same family living room at 6 SNRs of -6, -3, 0, 3, 6, 9 dB. Since our study focuses on monaural speech processing, only single channel signals (left ear) are used.

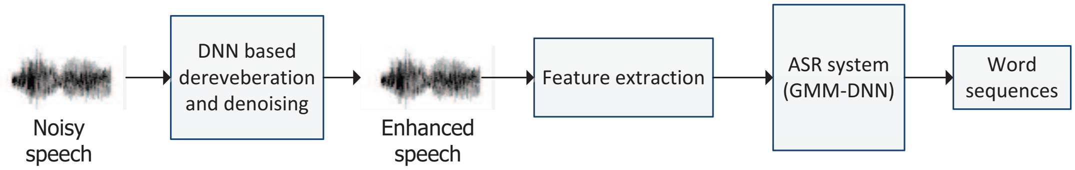


Figure 5.12: Diagram of an ASR system with a DNN based front-end for dereverberation and denoising.

To perform ASR, the proposed approach is treated as a front-end to enhance all sentences in both training and test datasets as shown in Fig. 5.12. We first randomly choose 3000 sentences from the CHiME-2 training set to train our DNN based dereverberation and denoising model. With the trained DNN model, we perform dereverberation and denoising for all sentences in the CHiME-2 training and test datasets, and resynthesize time-domain signals to construct new training and test datasets. No post-processing is used in this experiment. We then train ASRs model using the new training set containing only processed sentences, and test the ASR model using the new test set. The baselines are ASRs model trained and tested using original sentences including both clean and reverberant noisy sentences in the CHiME-2 corpus.

We use Kaldi toolkit [101] to train two ASR systems, each of which is trained using original sentences and processed sentences, respectively. The first ASR system is a standard GMM-HMM based system using MFCC features with triphone three-state model. Speaker adaptive training [2] is performed during the training stage. Another system is a hybrid ASR system, which uses alignments achieved from the GMM-HMM system and then trains DNNs with Mel-frequency filter bank features. This training scheme is motivated by [123], which achieves excellent performance on the CHiME-2 corpus. Sequence training [71] is also incorporated into this system.

We evaluate ASR performance in terms of word error rates (WERs). As shown in Fig. 5.13, for both GMM and hybrid ASR systems, the systems trained on processed sentences achieve lower WERs than those trained on original sentences across all SNR conditions. DNN based dereverberation and denoising considerably boosts ASR performance in low SNRs, where the improvements are 11.7% (absolute) for the GMM system and 3.3% for the hybrid system in -6 dB SNR. The advantage

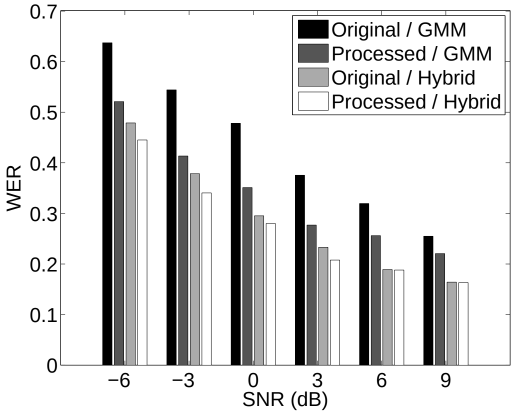


Figure 5.13: ASR results. “Original / GMM” and “Processed / GMM”denote the results for the GMM-HMM systems using original sentences and processed sentences, respectively. “Original / Hybrid” and “Processed / Hybrid” denote the results for the hybrid systems using original sentences and processed sentences, respectively.

gradually decreases as the SNR increases, because the performance decrement caused by reverberation and noise becomes smaller. On average, the improvements from original sentences are 9.5% for the GMM system and 2.0% for the hybrid system, demonstrating that our approach can be used as a front-end to improve ASR performance. We mention that the ASR experiments aim to show the advantage of the DNN based dereverberation and denoising rather than reaching the state-of-the-art results, which can be achieved in [91].

5.5 Discussion

We have proposed a supervised learning approach to perform dereverberation and denoising. The DNN is trained to learn a spectral mapping between corrupted speech and clean speech. Since temporal dynamics provides rich information for speech, the feature in this study is a concatenation of spectral features in a window. A more fundamental approach to utilize temporal information is to use an RNN, which is a natural extension of a feedforward network. An RNN aims to capture long-term temporal dynamics using time-delayed self-connections and is trained sequentially. We have trained RNN models for spectral mapping, and yielded around 0.2 dB improvement in terms of SNR_{fw}. Although this improvement is not significant, it is worth exploring RNNs in future work, for example, long short-term memory (LSTM) [51].

In our experiment, we train the DNN model using the IEEE corpus, which includes only one speaker. In order to test speaker dependency, we have also conducted experiments using the TIMIT corpus [151], where multiple speakers, including both male and female, are contained in the training dataset. We have tested the model for new speakers and achieved similar performance as that with the IEEE corpus. Note that, in our ASR experiments in Section 5.4.4 using the CHiME-2 corpus training and testing were conducted in a speaker-independent manner, showing that our approach is robust to different speakers.

It is worth mentioning that we have attempted to train a DNN based mapping on the cochleagram using the gammatone filterbank [134]. In this case, an element of an input vector corresponds to the log energy of each T-F unit of corrupted speech, while that of an output vector corresponds to the log energy of each T-F unit of clean speech. The DNN based cochleagram mapping also produces accurate cochleagram estimates, and the results are comparable with the spectrogram mapping.

In our ASR experiments, we resynthesize time-domain signals from DNN outputs and then perform speech recognition based on processed signals. According to our experiments, although the iterative signal reconstruction improves predicted speech intelligibility and quality scores, it does not lead to significant improvement for ASR performance. Comparing Fig. 5.2(c) with Fig. 5.2(e), the DNN output is still better than the spectrogram of the reconstructed signal, suggesting that we may extract MFCC or Mel filterbank features directly from the DNN output without resynthesis. As the DNN output is a better spectral representation than the spectrogram of resynthesized signals, we expect it can yield better ASR performance. This should be explored in a future work.

To sum, we have proposed to use DNNs to learn a spectral mapping from corrupted speech to clean speech for dereverberation, and dereverberation plus denoising. To our knowledge, this is the first study employing supervised learning for the problem of speech dereverberation. This novel approach is conceptually simple. Our supervised learning approach significantly improves dereverberation, as well as denoising, performance in terms of predicted speech intelligibility and quality scores, and boosts ASR results in a range of reverberant and noisy conditions.

CHAPTER 6

NEURAL NETWORK BASED PITCH TRACKING

Pitch determination is a fundamental problem in speech processing, which has been studied for decades. However, it is challenging to determinate pitch in strong noise because the harmonic structure is corrupted. In this chapter, we estimate pitch using supervised learning, where the probabilistic pitch states are directly learned from noisy speech data. We investigate two alternative neural networks modeling pitch state distribution given observations. The first one is a DNN, which is trained on static frame-level acoustic features. The second one is a RNN which is trained on sequential frame-level features and capable of learning temporal dynamics. Both DNNs and RNNs produce accurate probabilistic outputs of pitch states, which are then connected into pitch contours by Viterbi decoding. Our systematic evaluation shows that the proposed pitch tracking algorithms are robust to different noise conditions and can even be applied to reverberant speech. The proposed approach also significantly outperforms other state-of-the-art pitch tracking algorithms. A preliminary version presented in this chapter has been published in the *Proceedings of the 2014 IEEE International Conference on Acoustic, Speech, and Signal Processing* [42]. We have also submitted a manuscript to *IEEE Transactions on Audio, Speech, and Language Processing*.

6.1 Introduction

Pitch, or fundamental frequency ($F0$), is one of the most important characteristics of speech signals. A pitch tracking algorithm robust to background interference is critical to many applications, including speaker identification [4] and speech separation [37]. Although pitch tracking has been studied for decades, it is still challenging to estimate pitch from speech in the presence of strong noise, where the harmonic structure of speech is severely corrupted.

A typical pitch determination algorithm consists of two stages. The first stage determines pitch candidates or computes the pitch probability for each time frequency unit. To deal with noise, previous studies either utilize signal processing to attenuate noise [32], [21] or employ statistical methods to model the harmonic structure [147], [18], [60]. However, the selection of pitch candidates is often ad hoc, and it may be less optimal to make a hard decision for pitch candidate selection. Statistical modeling usually relies on strong assumptions, which make the algorithms difficult to generalize to complex acoustic environments. In the second stage, the pitch candidates or probabilities are connected into pitch contours using dynamic programming [32], [18] or hidden Markov models (HMMs) [147], [65].

It is sensible to formulate the pitch determination problem as an HMM decoding problem, where a hidden state corresponds to a pitch frequency and an observation corresponds to acoustic features. This way, pitch determination is equivalent to finding the optimal sequence of hidden states given an observation sequence. In an HMM, a key problem is to estimate the posterior probability given the observation in each time step. In this study, we propose to supervisedly learn the posterior probability that a frequency bin is pitched given the observation.

A DNN is a feed-forward neural network with more than one hidden layer [49], which has been successfully used in signal processing applications [88, 140]. In automatic speech recognition, the posterior probability of each phoneme state is modeled by a DNN. We adopt this idea for pitch tracking, i.e., we use a DNN to model the posterior probability of each pitch state given the observation in each frame. The DNN is expected to generate accurate probabilistic outputs due to its powerful learning capacity.

Further, speech has prominent temporal dependency which provides rich information for speech processing. A straightforward method to capture temporal information is to include neighboring frames into an expanded feature vector. However, this technique can only capture the temporal information within a limited span, because the dimensionality of the feature is proportional to the number of the frames and it is difficult to train a model with very high dimensional features. To utilize temporal dynamics, a more systematic approach is to directly encode temporal information into learning machines. A RNN is an extension of the feedforward neural network, where the hidden units have delayed self-connections. These recurrent connections allow the network to encode temporal information suitable for modeling nonlinear dynamics. Recent studies have shown promising results using RNNs to model sequential data [131], [85]. Given that speech is inherently a sequential signal and temporal dynamics is crucial to pitch tracking, we consider RNNs to model the probability distribution of pitch states.

To recapitulate, we investigate DNN and RNN based supervised methods for pitch tracking in very noisy speech. With proper training, both DNN and RNN are expected to produce reasonably accurate probabilistic outputs for pitch states. With the pitch state probability in each frame, a Viterbi decoding algorithm will be utilized to form continuous pitch contours (see also [147]).

This chapter is organized as follows. The next section relates our work to previous studies. Section 6.3 discusses the feature extraction part. The details of the proposed pitch tracking approach are presented in Section 6.4. The experimental results and comparisons are presented in Section 6.5. We discuss related issues and conclude the chapter in Section 6.6.

6.2 Related prior work

Recent studies on robust pitch tracking have explored either harmonic structure in the frequency domain, periodicity in the time domain, or the periodicity of individual frequency subbands in the time-frequency domain.

In the frequency domain, harmonic structure exhibits rich information about pitch. Previous studies extract pitch from the spectrum of speech, by assuming that each peak in the spectrum corresponding to a potential harmonic [117], [47]. SAFE [18] utilizes prominent SNR peaks in speech spectra to model the distribution of pitch using a probabilistic framework. PEFAC [32] combines nonlinear amplitude compression to attenuate narrowband noise and chooses pitch candidates from the filtered spectrum.

Another type of approaches utilizes the periodicity of speech in the time domain. RAPT [124] calculates the normalized ACF and chooses the peaks as the pitch candidates. The YIN [21] algorithm uses the squared difference function based on ACF to identify pitch candidates.

An extension of time-domain approaches extracts pitch using the periodicity of individual subbands in the time-frequency domain. Wu *et al.* [147] model pitch period statistics on top of a channel selection mechanism and use an HMM for extracting continuous pitch contours. Jin and Wang [65] use cross-channel correlation to select reliable channels and derive pitch scores from resulting summary correlogram. Huang and Lee [60] compute a temporally accumulated peak spectrum to estimate pitch. Lee and Ellis [77] extract the ACF features and train an MLP classifier on the principal components of the ACF features for pitch detection.

Different from the above methods, we use spectral domain features to provide a robust representation for pitch tracking in noise. Further, our approach utilizes advanced classifiers, namely deep neural networks and recurrent neural networks, which generate accurate probabilistic pitch states and boost the pitch tracking performance.

In addition, we believe that a large dataset with multiple conditions benefits robustness of the proposed algorithms to noises and reverberation.

6.3 Feature extraction

The proposed pitch tracking algorithms first extract spectral domain features in each frame, and then employ neural networks to compute the posterior probability of the pitch state for each frequency bin. With probabilistic outputs, we use Viterbi decoding to connect pitch states and form final pitch contours.

The features used in this study are extracted from the spectral domain based on [32]. We compute the log-frequency power spectrogram and then normalize to the long-term speech spectrum to attenuate noises. A filter is then used to enhance the harmonicity.

Specifically, a signal is first decomposed to the spectral domain using short time Fourier transformation. Let $X_t(f)$ denote the power spectral density (PSD) of the frame t in the frequency bin f. The PSD in the log-frequency domain can be represented as $X_t(q)$, where $q = \log f$. Then, the normalized PSD can be computed as:

$$X'_t(q) = X_t(q)\frac{L(q)}{\overline{X}_t(q)} \tag{6.1}$$

where $L(q)$ represents the long-term average speech spectrum, and $\overline{X}_t(q)$ denotes the smoothed averaged spectrum of speech, which is calculated by using a 21-point moving average filter in the log-frequency domain and averaging over the entire sentence (2∼4 s duration) in the time domain in this study.

With the normalized spectrum, we further enhance harmonicity for pitch tracking using a filter with broadened peaks having an impulse response defined as:

$$h(q) = \begin{cases} \dfrac{1}{\gamma - \cos(2\pi e^q)} - \beta, \text{ if } \log(0.5) < q < \log(K+0.5) \\ 0, \text{otherwise} \end{cases} \tag{6.2}$$

where β is chosen so that $\int h(q)dq = 0$, and γ controls the peak width which is set to 1.8.

The convolution $\tilde{X}_t(q) = X'_t(q) \star h(q)$ contains peaks corresponding to harmonics and their multiples and submultiples. Only the spectral components in the plausible pitch frequency range (60 to 400 Hz in this study) are selected as features. So we have a spectral feature vector in frame t:

$$\tilde{\mathbf{x}}_t = (\tilde{X}_t(q_1), \ldots, \tilde{X}_t(q_n))^T$$

Gonzalez and Brookes [32] proposed to extract the spectral feature $\tilde{\mathbf{x}}_t$ for pitch tracking in noise. Ideally, the pitch, $F0$, can be found by taking the highest peak in $\tilde{\mathbf{x}}_t$. In [32], several highest peaks are chosen for each frame as pitch candidates, and a dynamic programming algorithm is then used to form pitch contours. Although the feature vector is designed to deal with noisy speech, rule-based pitch candidate selection may lose useful information because it simply ignores non-peak spectral information. In our study, we treat $\tilde{\mathbf{x}}_t$ as the extracted feature and employ supervised learning to estimate pitch probability, i.e. to learn the mapping from the features to the pitch frequencies. We expect supervised learning to yield better results.

Since neighboring frames contain useful information for pitch tracking, we incorporate the neighboring frames into the feature vector. Therefore, the final frame-level feature vector is

$$\mathbf{x}_t = (\tilde{\mathbf{x}}_{t-d}, \ldots, \tilde{\mathbf{x}}_{t+d})^T$$

where d is set to 2 in our study.

6.4 Learning pitch state distribution

Instead of selecting pitch candidates, we employ supervised training approach to learn the posterior probability distribution given the features in each frame. Neural networks have recently achieved large progress in speech processing, and we propose to use two kinds of neural networks to model the probability distribution.

6.4.1 DNN based pitch state estimation

Our first method is to use a feedforward DNN. To simplify the computation, we quantize the plausible pitch frequency range into M frequency bins, corresponding to M pitch states $s^1, \ldots, s^M$. Also, we incorporate a nonpitch state s^0 corresponding to an unvoiced speech or speech-free state. We use 24 bins per octave in a logarithmic scale to quantize the plausible pitch frequency range into 67 bins. So there are totally 68 states [77].

To train the DNN, each training sample is the feature vector $\mathbf{x}_t$ in the time frame t (and its neighboring frames), and the target is an $(M + 1)$-dimensional vector of the pitch states $\mathbf{s}_t$, whose element s_t^i is 1 if the groundtruth pitch falls into the corresponding frequency bin, and 0 otherwise.

In order to learn the probabilistic output, we use cross-entropy as the objective function.

$$\mathcal{L}(\mathbf{y}, \mathbf{x}; \mathbf{\Theta}) = -\sum_{m=0}^{M} y_m \ln f_m(\mathbf{x}) \tag{6.3}$$

where $\mathbf{y} = (y_0, \ldots, y_M)^T$ is the desired output and $f_m(\cdot)$ is the actual output of the mth neuron in the output layer. $\mathbf{\Theta}$ denotes the parameters we need to learn. The activation function in the hidden layers is the sigmoid function and the output layer uses the softmax function for probabilistic outputs.

The DNN in this study includes three hidden layers with 1600 sigmoid units in each layer, and a softmax output layer whose size is set to the number of the pitch states, i.e., 68 output units. The number of hidden layers and the hidden units are chosen from cross validation (see also Sect. V.B). We use backpropagation with mini-batch stochastic gradient descent to train the DNN model, and the actual cost in each mini-batch is computed from the summation over multiple training samples using Eq. 6.3.

The trained DNN produces the posterior probability of each pitch state i: $P(s_t^i|\mathbf{x}_t)$.

6.4.2 RNN based pitch state estimation

The DNN based method utilizes frame-level features to compute the posterior probabilities of pitch states. Although it utilizes neighboring frames to incorporate temporal information, it is not able to capture long-term temporal dynamics due to the limit of feature dimensionality. As temporal continuity and variation are important characteristics of pitch, we explore a more intrinsic method to capture temporal context information.

An RNN is a natural extension of a feedforward network. In an RNN, the depth comes from not only multiple hidden layers but also unfolding layers through time. An RNN is capable of capturing the long-term dependencies through connections between hidden layers. These attributes have inspired us to use RNNs to model pitch dynamics. One of the key challenges for using RNNs is that training with long-term dependencies can be quite difficult and some new approaches have been proposed to address the problem [121]. In our study, we use a classic RNN [26] and learn the model with truncated backpropagation through time (BPTT) [112], [142].

The RNN has hidden units with delayed connections to themselves, and the output $\mathbf{y}_o = (y_1, \dots, y_n)^T$ of the RNN at the time step t can be represented as:

$$\begin{aligned}
&\mathbf{y}_o(t) = \psi(\mathbf{W}_{o,j}^T \mathbf{h}_j(t)) \\
&\mathbf{h}_j(t) = \phi(\mathbf{v}_j(t)) \\
&\mathbf{v}_j(t) = \mathbf{W}_{j,j-1}^T \mathbf{h}_{j-1}(t) + \mathbf{W}_{j,j}^T \mathbf{h}_j(t-1) \\
&\mathbf{h}_1(t) = \phi(\mathbf{W}_{1,i}^T \mathbf{x}_i(t))
\end{aligned} \tag{6.4}$$

where ϕ and ψ are the sigmoid function and the softmax function respectively. $\mathbf{W}_{j,j-1}$ denotes the weights matrix from the $j-1$th hidden layer to the jth hidden layer, and the numbers of the rows and the columns are equal to the number of the units in the $j-1$th layer and the jth layer, respectively. $\mathbf{h}_j$ is a column vector corresponding to the activations of the jth hidden layer. $\mathbf{W}_{j,j}$ denotes the self-connections in the jth layer. Note that, since each unit only has a recurrent connection to itself, $\mathbf{W}_{j,j}$ is a diagonal matrix. For a non-recurrent hidden layer, $\mathbf{W}_{j,j} = \mathbf{0}$. $\mathbf{W}_{o,j}$ specifies the weight matrix between the last hidden layer and the output layer, and $\mathbf{W}_{1,i}$ the weight matrix between the input layer and the first hidden layer. For a recurrent hidden layer, the state of a neuron is influenced by not only the external input to the network but also the network activation from the previous time steps.

With recursion over time on hidden units, an RNN can be unfolded through time and can be viewed as a very deep network with T layers, where T is the number of time steps. The structure of the RNN in our study includes two hidden layers. Each hidden layer has 256 hidden units and only the units in the second hidden layer have self-connections. The input and the output layers are the same as in the DNN.

To use the truncated BPTT to train the RNN, each training sentence is truncated into multiple segments with a fixed length of T frames. Each segment is treated as a sequential training sample and fed into the neural network. To train the network, the RNN is unfolded for T time steps, and the backpropagated error $\delta_j(t)$ for a neuron in the recurrent layer j is computed from both the next layer $\delta_{j+1}(t)$ and the next time step $\delta_j(t+1)$. Although the truncated BPTT cannot capture the temporal information exceeding T time steps, the training is relatively easy. In our experiment, we set $T = 15$ and a longer T does not significantly improve the performance.

In the test phase, the output of the RNN is computed sequentially, and the output of the RNN in the tth frame is the posterior probability $P(s_t^i|\mathbf{x}_1, \ldots, \mathbf{x}_t)$, where the observation is a sequence from the past to the current frame instead of the feature $\mathbf{x}_t$ in the current frame.

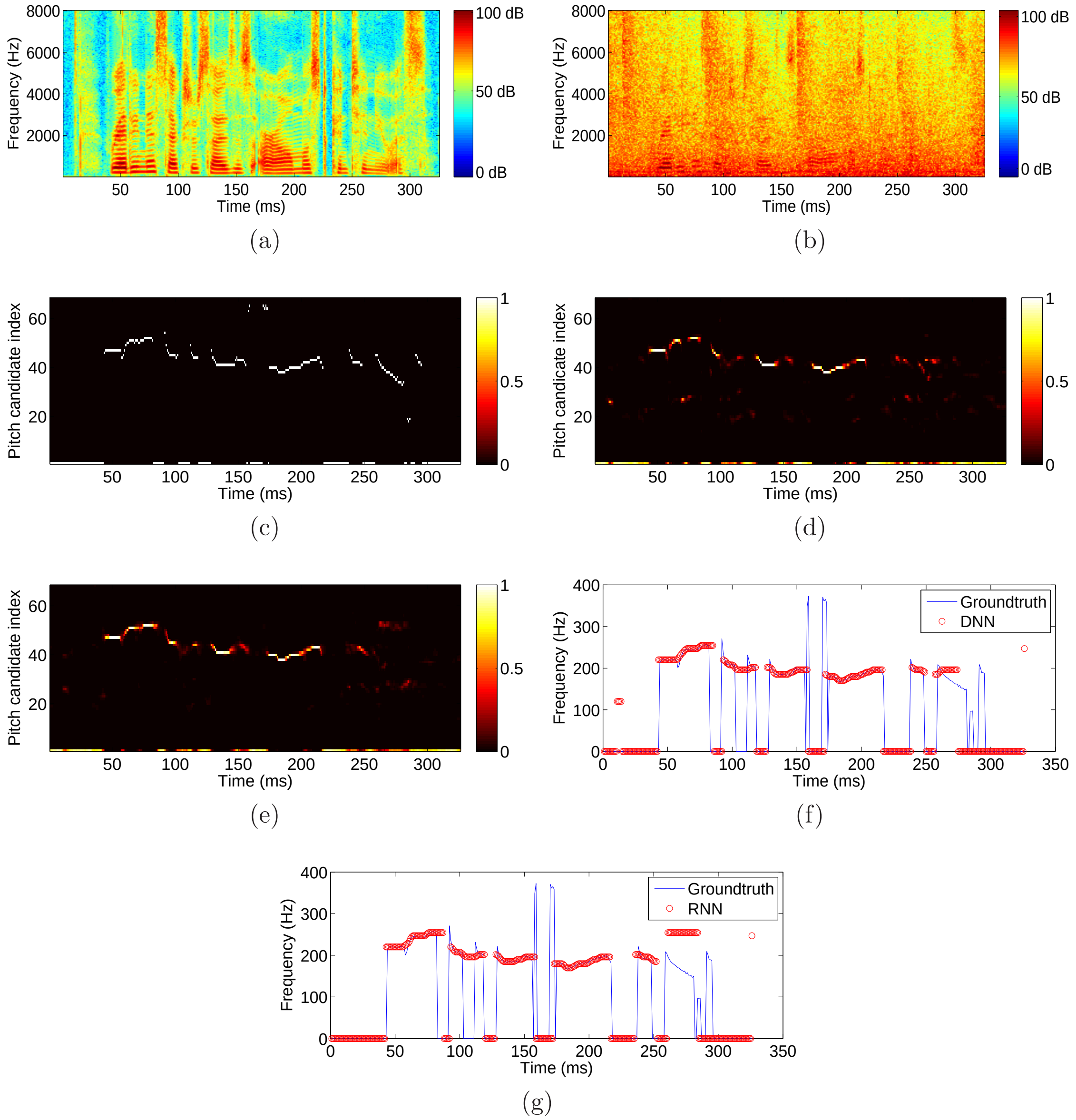
6.4.3 Viterbi decoding

The DNN or the RNN produces the posterior probability distribution in each time frame. We then use Viterbi decoding [29], [147] to connect those pitch states according to neural network outputs.

The Viterbi algorithm utilizes the likelihood and the transition probability to calculate the cost in order to generate an optimal sequence. The likelihood in each frame $P(\mathbf{x}_t|s_t^i)$ is proportional to the posterior probability divided by the prior $P(s^i)$:

$$p(\mathbf{x}_t|s_t^i) \propto \frac{P(s_t^i|\mathbf{x}_t)}{P(s^i)} \tag{6.5}$$

where $P(s_t^i|\mathbf{x}_t)$ is the output of a neural network. The prior $P(s^i)$ and the transition matrix are directly computed from the training data. Note that, since we train the DNN with both pitched and unpitched frames, the prior of the unpitched state $P(s^0)$ is usually much larger than that of each individual pitched state, resulting in the relatively small likelihood of the unpitched state, and the Viterbi algorithm may bias towards pitched states. Hence, we introduce a parameter $\alpha \in (0, 1]$ multiplying



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 6.1: (Color online) Neural network based pitch tracking. Noisy speech is a female utterance from the TIMIT corpus "Readiness exercises are almost continuous", mixed with factory noise in -5 dB SNR. (a) Spectrogram of clean speech. (b) Spectrogram of noisy speech. (c) Groundtruth pitch states. In each time frame, the probability of a pitch state is 1 if it corresponds to the groundtruth pitch and 0 otherwise. (d) Probabilistic outputs from the DNN. (e) Probabilistic outputs from the RNN. (f) DNN based pitch contours. The circles denote the generated pitches, and solid lines denote the groundtruth pitch. (g) RNN based pitch contours.

the prior of the unpitched state $P(s^0)$ to balance the ratio between the pitched and unpitched states, which is chosen from a development set. We should also mention that the output of the RNN is the posterior probability given an observation of a sequence rather than a single frame, which does not exactly satisfy the assumption of the HMM and the Viterbi algorithm, but we ignore this for simplicity.

The Viterbi algorithm outputs a sequence of pitch states for a sentence. We convert the sequence of pitch states to the sequence of frequencies and then use a 3-point moving average for smoothing to generate final pitch contours.

Fig. 6.1 illustrates pitch tracking results using the proposed methods. The example is a female utterance from the TIMIT corpus [151], "Readiness exercises are almost continuous", mixed with factory noise in -5 dB SNR. Fig. 6.1(a) and (b) show the spectrograms of clean speech and noisy speech respectively. Comparing Fig. 6.1(b) with Fig. 6.1(a), the harmonics are severely corrupted by noise, leading to a major difficulty in pitch tracking. Fig. 6.1(c) shows the groundtruth pitch states extracted from the clean speech using Praat [10]. As shown in the figure, Praat even makes a few doubling or halving pitch errors at around 160 ms and 280 ms, but since these errors are not serious, we do not correct them and still treat them as the groundtruth. The probabilistic outputs of the DNN and the RNN are shown in Figs. 6.1(d) and (e), respectively. Comparing to Fig. 6.1(c), the probabilities of the correct pitch states dominate in most time frames in both Figs. 6.1(d) and (e), demonstrating that the neural networks successfully predict pitch states from noisy speech. In some time frames (e.g., 100 ms to 120 ms), the RNN yields better probabilistic outputs than the DNN, because the RNN is able to better capture the temporal context and its outputs are smoother than those of the DNN. Figs. 6.1 (f) and (g) show pitch contours after Viterbi decoding. In the figures, both the DNN and the RNN produce accurate pitch contours. A few errors occur from 260 ms to 280 ms due to severe interference.

6.5 Experimental results

6.5.1 Corpus

We evaluate the performance for the proposed approach using the TIMIT database [151], [65]. The training set contains 250 utterances including 50 male speakers and 50 female speakers. The noises used in the training phase include babble noise from [52], factory noise, and high frequency radio noise from NOISEX-92 [128]. Each utterance is mixed with every noise type in three SNR levels: -5, 0, and 5 dB, therefore the training set includes 250 × 3 × 3 = 2250 noisy sentences. The test set contains 20 utterances including 10 male speakers and 10 female speakers. All utterances and speakers are not seen in the training set. The noise types used in the test set include the three training noise types and six new noise types: cocktail-party noise, crowd playground noise, crowd music, traffic noise, wind noise, and rain noise [53]. We point out that although the three training noises are included in the test set, the noise recordings are cut from different segments. Each test utterance is mixed with each noise in six SNR levels of -10, -5, 0, 5, 10, and 20 dB. The groundtruth pitch is extracted from clean speech using Praat [10]. In addition, we test the proposed approach using 20 utterances in the FDA evaluation database [7] where the groundtruth pitch contours were derived from laryngograph data.

We evaluate pitch tracking results in terms of two measurements: detection rate (DR) [57] and voicing decision error (VDE) [90]. DR is evaluated on voiced frames, where a pitch estimate is considered correct if the deviation of the estimated $F0$ is within 5% of the groundtruth $F0$, and VDE indicates the percentage of frames are misclassified in terms of voicing:

$$\mathrm{DR} = \frac{N_{0.05}}{N_p}, \qquad \mathrm{VDE} = \frac{N_{p \to n} + N_{n \to p}}{N} \tag{6.6}$$

Here, $N_{0.05}$ denotes the number of frames with pitch frequency deviation smaller than 5% of the groundtruth frequency. $N_{p \to n}$ and $N_{n \to p}$ denote the number of frames misclassified as unpitched and pitched, respectively. N_p and N are the number of pitched frames in groundtruth and total frames in a sentence, respectively.

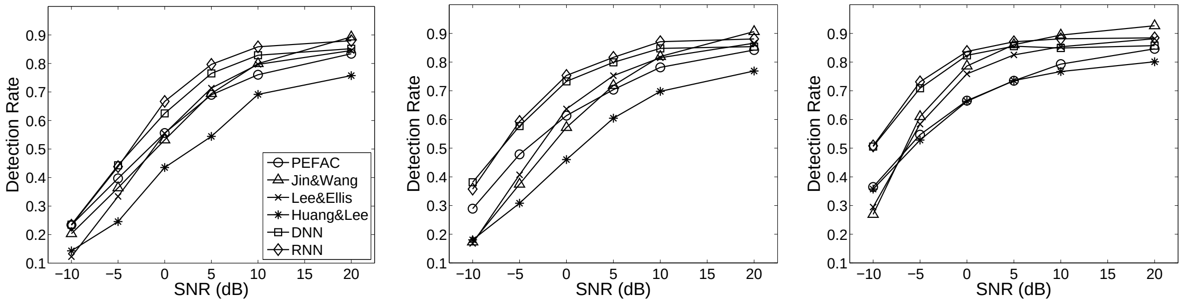
6.5.2 Parameter selection

Since the proposed neural networks involve several parameters, we describe how to choose their values in this subsection. The size of training set influences on the performance, and we train three DNN models using different training sets with 450, 1350, and 2250 noisy sentences. We compare the pitch tracking results on both training noise types and new noise types. On average, the selected training set with 2250 noisy sentences performs better than the one with 450 noisy sentences by around

6.6% in terms of detection rates, and slightly outperforms the one with 1350 noisy sentences.

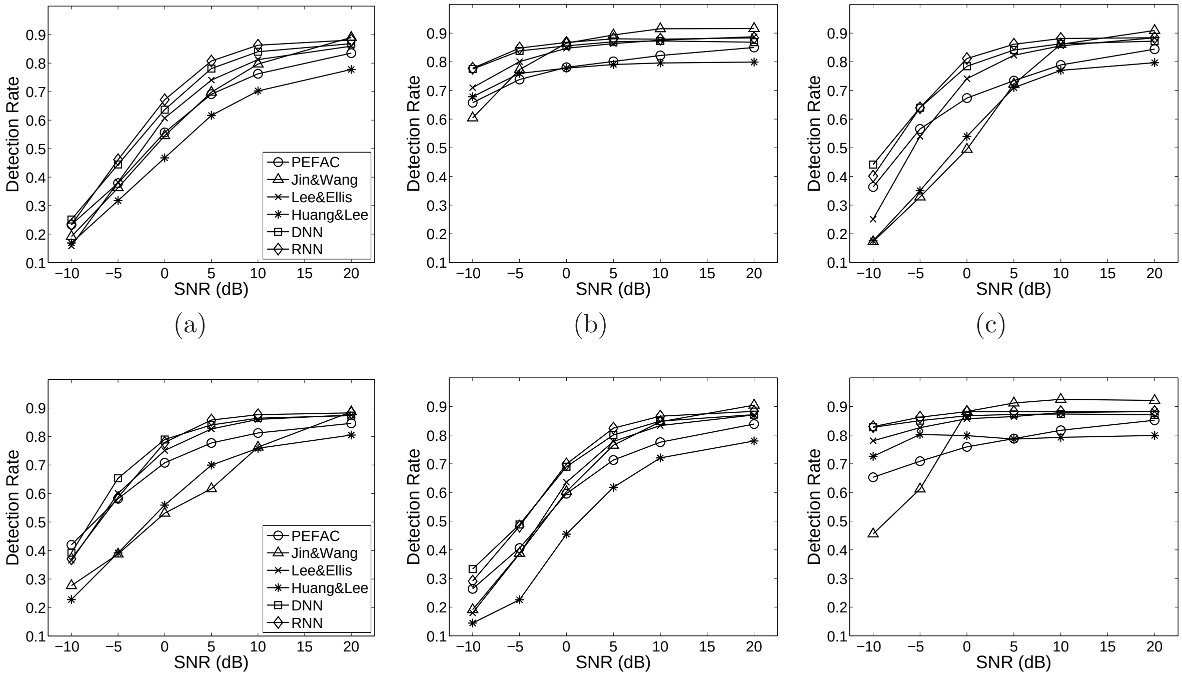
Another important factor concerns features. In this study, we first compute the PSD $X(q)$ in the log-frequency domain, and then generate the normalized PSD $X'(q)$. The normalized spectral features are then convolved with a filter with a broadened impulse response, resulting the final features used in our study $\tilde{X}(q)$. To reveal feature effects, we train three DNN models using different features. The experiments show that the filtered normalized PSD achieves the best performance, and the normalized PSD and the original PSD achieve comparable performance. The average detection rates are boosted by 5.0% for seen noises and 6.9% for unseen noises by using the filtered normalized PSD.

We have conducted experiments for both DNN and RNN using different numbers of hidden layers. In our experiments, the DNN with three hidden layers performs better than that with one hidden layer by 2.6% in detection rate and that with two hidden layers by 1.3%. The RNN with two hidden layers produces comparable performance to that with three hidden layers, but outperforms that with one hidden layer by 3.4%. We have also evaluated different numbers of hidden units, learning rates, and the numbers of neighboring frames. The parameter values used in this study are chosen using cross-validation from a development set.



(a) (b) (c)

Figure 6.2: Pitch detection rate comparisons for (a) babble noise, (b) factory noise, (c) high frequency radio noise.



(a) (b) (c)

(d) (e) (f)

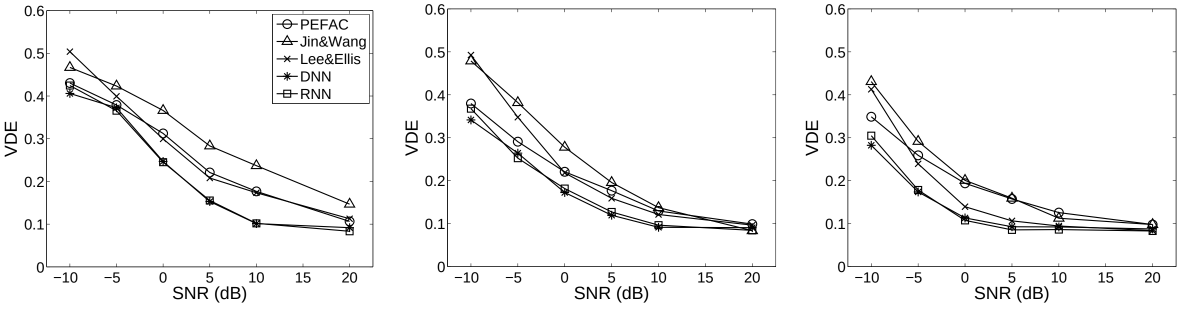
Figure 6.3: Pitch detection rate comparisons for six new noises: (a) cocktail-party noise, (b) crowd playground noise, (c) crowd music, (d) traffic noise, (e) wind noise, (f) rain noise.

6.5.3 Results and comparisons

We compare our approach with four pitch tracking algorithms. PEFAC [32] extracts normalized spectral features to deal with strong noise and produces competitive pitch tracking results. The multipitch tracking algorithm of Jin and Wang [65] computes the autocorrelation function to select reliable channels and then utilizes an HMM to generate pitch contours (see also [147]). This algorithm is designed to handle reverberant noisy conditions. The third algorithm was proposed by Huang and Lee [60]. They compute a temporally accumulated peak spectrum as features and apply sparse reconstruction to estimate pitch in noise. The fourth algorithm was proposed by Lee and Ellis [77]. They extract subband autocorrelation and apply principal component analysis to reduce dimensionality. They train an MLP to estimate pitch. Note that, like ours the latter two algorithms require training and we use the same corpus (see Section V.A) to train these models for comparison.

Fig. 6.2 shows the detection rates for three training noises. The detection rates gradually increase with the increase of SNR. The DNN and the RNN based methods achieve substantially higher detection rates than others, especially in very low SNR conditions. The results of the unsupervised PEFAC algorithm are also notable, particularly for babble noise. Although we do not train the models under the SNR of -10 dB, the proposed approach still outperforms the others in this very low SNR condition. For the untrained high SNR conditions, the proposed approach also achieves good performance, although the relative advantage to others is not as large as in low SNRs. The proposed approach performs more than 6% better than all others on average and the advantage is more than 10% when the SNR is below 5 dB. The RNN performs slightly better than the DNN when the SNRs are greater than -5 dB.

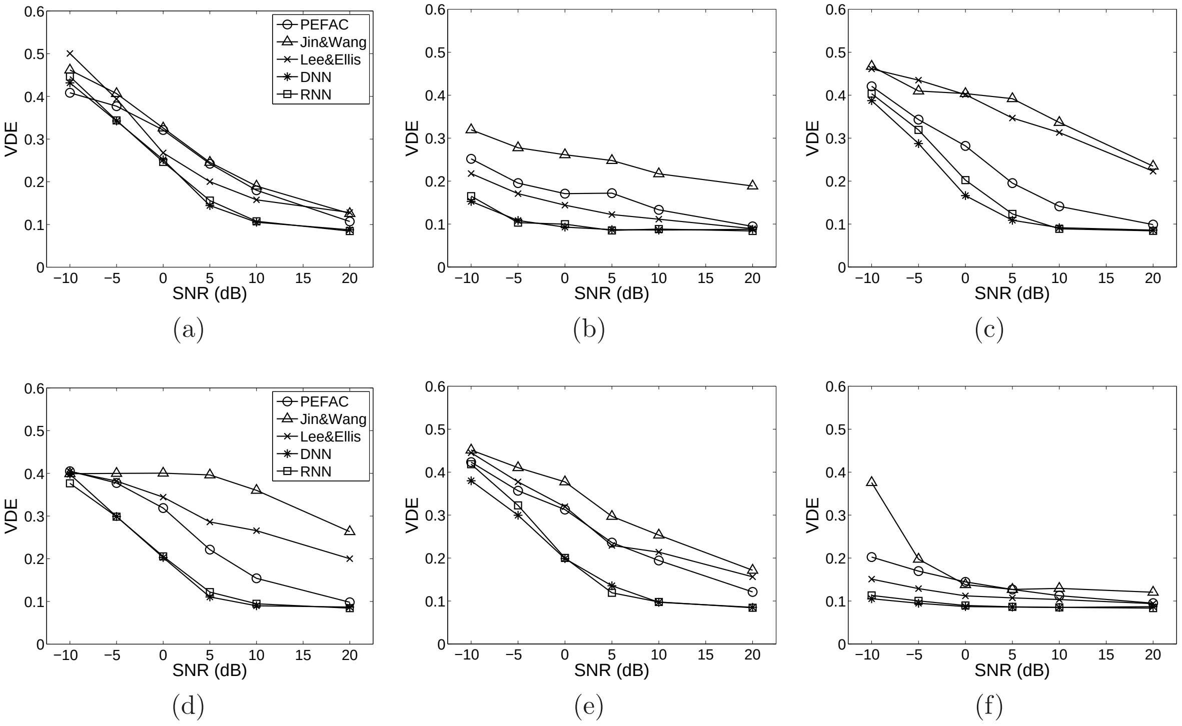
Fig. 6.3 shows the detection rates for six new noises that are not seen in the training phase. Similar to Fig. 6.2, this figure shows that the proposed approach yields the best performance in these noise conditions, demonstrating that our supervised learning algorithms generalize well to different noisy environments. The average detection rates for the DNN and the RNN are 75% and 76% respectively, while the best comparison result is 71% for Lee and Ellis.



(a) (b) (c)

Figure 6.4: Voicing detection error comparisons for (a) babble noise, (b) factory noise, (c) high frequency radio noise.

It is desirable for a pitch tracking algorithm to achieve high detection rates and low voicing detection errors at the same time. Since Huang and Lee's algorithm does not produce a pitched/unpitched decision, we only compare our approach with PEFAC,



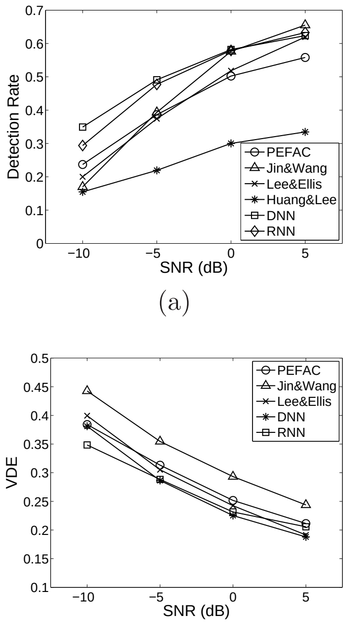
(a) (b) (c)

(d) (e) (f)

Figure 6.5: Voicing detection error comparisons for six new noises: (a) cocktail-party noise, (b) crowd playground noise, (c) crowd music, (d) traffic noise, (e) wind noise, (f) rain noise.

Jin and Wang, and Lee and Ellis. Fig. 6.4 and Fig. 6.5 show the VDE results for the seen and unseen noises, respectively. As shown in the figures, our algorithms produce lower voicing detection errors than others. On average, the VDEs of the DNN and the RNN based methods are 17% and 18% across 3 SNRs and 6 noises, and the VDEs of PEFAC, Jin and Wang, and Lee and Ellis are 23%, 28%, and 24%, respectively. The results indicate the superiority of the proposed approach on both pitch and voicing detection.

In the above experiments, the groundtruth pitch is extracted from clean speech using Praat, which is prone to some pitch detection errors. We now use the FDA database [7] to evaluate our approach without any retraining, where the groundtruth pitch is derived from laryngograph data. Fig. 6.6 shows the average pitch tracking results over three training noises and four different SNRs. The average detection rates for the DNN and the RNN are 51% and 50% respectively, which are higher than the others by around 6%. These and voicing detection results are consistent with those using Praat detected pitch as groundtruth.



(a)

(b)

Figure 6.6: Pitch tracking results for the FDA database. (a) Pitch detection rate. (b) Voicing detection error.

In Eq. 6.6, the denominator of the detection rate is the number of all pitched frames in the groundtruth, so it counts false rejects as errors. Other studies [147], [90] used gross pitch error (GPE) to evaluate pitch deviation over 20% in the frames where both the groundtruth and a pitch estimator produce a pitch. We have also used GPE to compare the performances of different approaches in six SNR conditions. The DNN and the RNN achieve GPEs of 6.6% and 5.7%, respectively. Lee and Ellis also achieve GPE of 5.7%. All others have GPEs higher than 9%.

VDE aggregates false rejects and false alarms together. Specifically, false reject is the percent of unpitched frames in a reference sentence wrongly classified by an estimator, and false alarm is the percent of pitched frames wrongly classified. Looking at these two kinds of error separately, the DNN and the RNN achieve low false reject rates in low SNR levels, that is, they can correctly detect pitched frames even when noise is very strong. On average, the false reject rates for the DNN and the RNN are 12% and 10% respectively, and Jin and Wang achieve the next best at 15%. The false alarm rates for all methods are comparable, below 7% under most conditions.

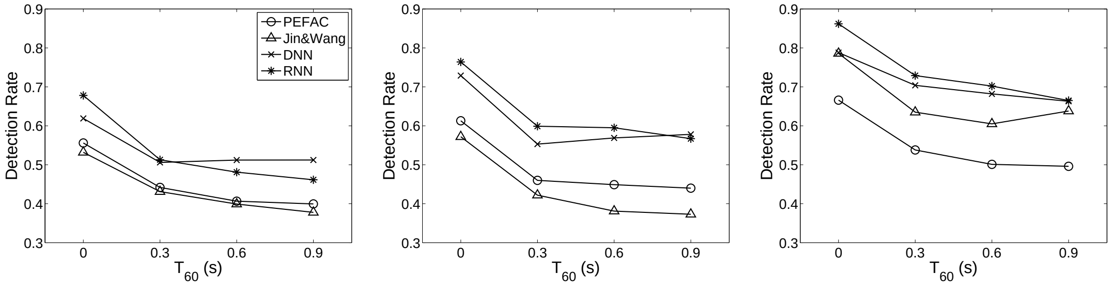
In terms of computational complexity, the processing time of the proposed approach is comparable with the other approaches. Most approaches take less than 2 seconds to process a one-second noisy speech signal, except for Jin and Wang which takes significantly more time.

6.5.4 Extension to reverberant conditions

Reverberation smears the characteristics of harmonic structure and thus makes the task of pitch tracking more difficult. We apply the proposed approach to reverberant and noisy speech to evaluate the performance. In voiced speech, the fundamental frequency is defined as the rate of vibration of the vocal folds [67]. However, in reverberant conditions, the received speech is the filtered aggregated signal and the actual periodicity of the reverberant speech does not necessarily match its anechoic version. Because some speech processing applications would prefer a pitch estimate consistent with the harmonic structure of the reverberant speech [63], we consider the pitch of the reverberant speech as the groundtruth (see [65]).

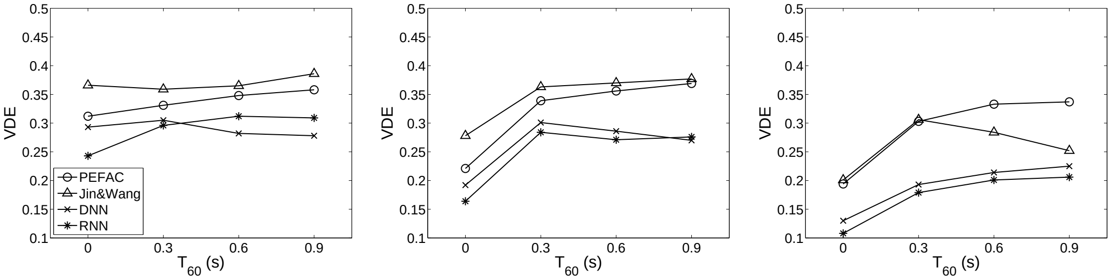
Because the groundtruth of reverberant speech is different from that of anechoic speech, we need to retrain the models in reverberant conditions. To simulate room acoustics, we generate a simulated room corresponding to a specific reverberation time T_{60} [35] and randomly create a set of room impulse responses (RIRs) under this T_{60} condition. To train the system, we generate three reverberation times: 0.3, 0.6, and 0.9 s. The training set includes 250 utterances and three noises, both of which are the same as in the previous subsection. For each T_{60} condition, an utterance and a noise signal are convolved with two different RIRs respectively, corresponding to different source locations, and the two reverberant signals are then mixed at 0 dB SNR. Therefore, there are $250 \times 3 \times 3 = 2250$ reverberant sentences in the training set. The test set includes 450 sentences, consisting of 50 utterances mixed with the three training noises in three T_{60}s. Although the three T_{60}s are used in the training set, the RIRs in the test set are different from those in the training set. The groundtruth pitch is extracted from reverberant and noise-free utterances using Praat.

We compare our approach with PEFAC and Jin and Wang, because both have been shown to perform well in reverberation. In Fig. 6.7 and Fig. 6.8, we present



(a) (b) (c)

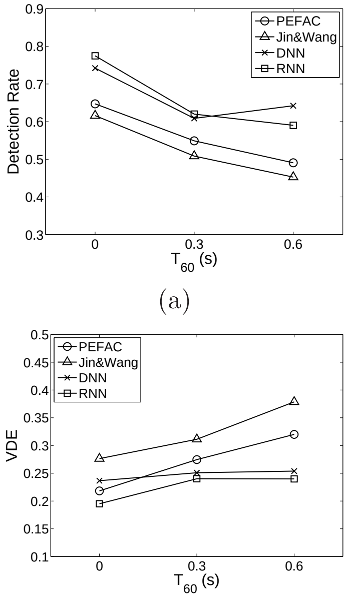
Figure 6.7: Pitch detection rates for reverberant noisy speech: (a) babble noise, (b) factoroy noise, (c) high frequency radio noise.



(a) (b) (c)

Figure 6.8: Voicing detection errors for reverberant noisy speech: (a) babble noise, (b) factoroy noise, (c) high frequency radio noise.

the DR and the VDE results for reverberant and noisy speech with three T_{60}s and anechoic speech. As shown in the figures, although the performance for noisy reverberant speech is lower than that in the anechoic condition, the increase of the reverberation time starting from 0.3 s does not lead to significant performance degradation. We should point out that the anechoic condition is an unseen condition in this experiment, because the retrained model only uses reverberant speech. The fact that these results are broadly comparable to those in Fig. 6.2 and Fig. 6.4 at 0 dB indicates insensitivity of our supervised approach to reverberation. The proposed approach performs substantially better than PEFAC and Jin and Wang in terms of both detection rates and voicing detection errors. Here, the RNN outperforms the DNN except for the high T_{60} conditions in the babble noise.



(a)

(b)

Figure 6.9: Pitch tracking results on an interactively labeled pitch corpus: (a) detection rate, (b) voicing detection error.

The above experiments use Praat to extract pitch from reverberant, noise-free speech as the groundtruth. As done in the previous subsection, we evaluate the approaches using another pitch evaluation corpus [66] where reference pitch contours are labeled from reverberant speech by an interactive pitch determination algorithm [86], combining automatic pitch determination and human intervention. The original sentences in the corpus are randomly selected from the TIMIT corpus. Each anechoic sentence is convolved with RIRs in $T_{60} = 0.3$ s and $T_{60} = 0.6$ s, respectively (see [65] for details). We generate reverberant and noisy signals using babble, factory, and high frequency radio noises at 0 dB SNR, and obtain pitch tracking results without retraining.

Fig. 6.9 gives the pitch and voicing detection results of our approach and those of the comparison methods. As shown in the figure, both the DNN and the RNN based algorithms lead to significantly higher detection rates for all three noises. On average, the detection rates for the DNN and the RNN are 66.4% and 66.2%, respectively; while those of the others are all below 57%. In terms of voicing detection errors, the proposed approach achieves the lowest error rate on average. Broadly speaking, these results show similar trends to those in Figs. 6.7 and 6.8, and hence suggest that it is reasonable to use Praat to generate groundtruth pitch for training.

6.6 Discussion

In this study, we use the supervised learning approach to learn the probability distribution of pitch states. Although supervised learning typically has a generalization issue, our system appears to exhibit very promising results across multiple unseen conditions, including different speakers, SNRs, noises, and room impulse responses. Some of previous supervised learning based pitch tracking algorithms perform well on trained conditions but need to be retrained in a new acoustic environment [18], [60]. We incorporate multiple conditions into a larger dataset and train a DNN or RNN model under different conditions, which potentially benefits the generalization ability of the system (see also [140]). The success of this multiple condition training is probably due to extracted robust features as well as the learning capacity of the neural networks. We have tried to train single condition models for each specific acoustic environment, and found that single-condition training performs only slightly better than our multi-conditions training.

Our acoustic features for pitch estimation are computed from the filtered normalized log-frequency power spectrogram. The features use signal processing techniques to attenuate interference and facilitate subsequent neural network training. We have attempted to add an ACF based feature [77], but it only yields a slight improvement. In principle, the DNN is capable of learning high-level representation from raw data [49], [8] and recent advances in speech recognition [79], [22] also demonstrate that a DNN directly trained on the filter-bank features achieves better performance than trained on MFCC features. This suggests that, instead of using signal processing to generate features, we may consider raw features for neural network training in the future.

We have trained both DNN and RNN for pitch state estimation. Since post-processing can correct some pitch estimation errors from neural network outputs, the RNN does not produce significantly better results than the DNN in some conditions. However, the RNN intrinsically captures temporal dynamics, making it well suited for pitch tracking. As an example, Figs. 6.1(d) and (e) show the difference in pitch state estimation by the DNN and the RNN, and we can see that the output of the

RNN is more smooth and continuous. In this study, we use the truncated BPTT to train the RNN and the longest time step is set to $T = 15$. A 15-frame truncation is not a long segment for pitch tracking, as the pitch contours in our study usually have 30 to 50 frames. We have tried to use 20-frame BPTT to train the models, but the results are similar, probably because training has reached a saturation point on our training dataset. Another strategy to train the RNN is to use BPTT on each sequence rather than a fixed-length segment. With sufficient training data the RNN is expected to encode longer dynamics, which may lead to performance improvement. In addition, we use a simple RNN in our study, and it is worth exploring other RNNs in future work, for example, LSTM [51], which has demonstrated better performance than the simple RNN in some applications [33].

With neural network outputs, we use the Viterbi algorithm to generate pitch contours in the framework of HMMs. In other words, we assume that 1) the observation only depends on the hidden state in the current time step, and 2) the hidden state in the current time step only depends on the previous hidden state. To relax these assumptions, some studies use conditional random fields (CRFs) to model the sequence [75], [30]. We have attempted to use the CRF to generate the best sequence, but the performance is only slightly better than Viterbi decoding. It may be because the neural networks yield adequate information and a simple post-processing technique can achieve good results. Due to its complexity, we do not incorporate the CRF in our system, but it will be interesting to explore better sequence models for pitch tracking.

To conclude, we have proposed DNN and RNN to estimate the posterior probabilities of pitch states for pitch tracking in highly noisy speech. The supervised learning based approach produces strong pitch tracking results in both seen and unseen noisy conditions. In addition, the proposed approach can be extended to reverberant conditions.

CHAPTER 7

CONCLUSION

7.1 Contributions

Monaural speech separation and processing are very challenging, and this dissertation addresses the problems through supervised learning. Most existing monaural speech separation and processing algorithms either employ signal processing and make assumptions about the statistical characteristics of signals, or train prior speech or noise models to reconstruct sound sources. In contrast, our approach is a data-driven approach, i.e., we use existing data to train models to capture the relationship between the noisy data and the target, and then use the trained models to predict the speech information in the test stage. Our novel supervised approach achieves promising results in many situations. We summarize our contributions in this section.

In Chapter 2, we present classification based speech separation to estimate the IBM. With effective features and powerful classifiers, our approach significantly boosts the classification performance. As demonstrated in [80] and [70], HIT−FA rates are well correlated with speech intelligibility. Since our system achieves higher HIT−FA rates than Kim *et al.*'s system, it is reasonable to expect that our system can lead to improved intelligibility.

Generalization is a major issue for supervised learning. If the training set and the test set have different distribution properties, one cannot expect a trained model to perform well in unmatched conditions. We address this problem in Chapter 3. We propose to use rethresholding to adaptively adjust the decision boundaries of SVMs, which is expected to generalize to new SNR or noise conditions. Our generalization approach can be directly applied to trained models, and does not need prior information for unseen conditions. Systematic evaluation and comparison show that the proposed approach produces high quality IBM estimates under unseen conditions.

In Chapter 4, we revisit the generalization problem in a more fundamental way. We utilize a metric learning approach for feature transformation, and the new features are robust to different noisy conditions. With the learned features, the speech separation system can be applied to new noise conditions. Evaluations show the advantage of the proposed approach over other speech separation systems.

Chapter 5 develops a DNN based dereverberation and denoising system. We propose to use DNNs to learn spectral mapping from reverberant and noisy speech to clean speech. To our knowledge, this is the first study using supervised learning for speech dereverberation. This novel approach is simple and effective at the same time. Our supervised learning approach significantly improves dereverberation and denoising performance, and boosts ASR results in a range of reverberant and noisy conditions.

In Chapter 6, we propose DNNs and RNNs to estimate the posterior probabilities of pitch states for pitch tracking in highly noisy speech. The supervised learning based approach produces accurate pitch tracking results in both seen and unseen noisy conditions. In addition, the proposed approach can be extended to reverberant situations. Our systematic evaluation shows that the proposed pitch tracking algorithms are robust to different noise and reverberation conditions.

In this dissertation, we first proposed the binary masking based approach for speech separation and then proposed the spectral mapping based approach for speech dereverberation. In fact, we have attempted to use binary masking based approach for speech dereverberation and the performance is slightly worse than spectral mapping approach. On the other hand, spectral mapping can be used to deal with speech separation when the noise is mild. However, if the noise is too strong, e.g., SNR is lower than 0 dB, the mapper may not be able to learn meaningful representation from the spectrum, mainly because the speech spectrum is severely corrupted by noise and the pattern of noisy speech is hard to be learned.

7.2 Future work

In this dissertation, we formulate speech separation and processing as a supervised learning problem, leading to a data driven approach. With the IBM considered as a computational goal for speech separation, the problem can be converted to binary classification. Although the IBM is the optimal binary mask, it may not necessarily be the most suitable target for training and prediction. IBM based speech separation has been shown to yield large speech intelligibility improvements, but it may not be as suitable for speech quality improvements. Other targets may need to be considered for training to improve both speech intelligibility and quality, for example, the ideal ratio mask, the magnitude spectrogram of clean speech, or the cochleagram of clean speech. As shown in Chapter 6, we have used the magnitude spectrogram as the training target and yielded good performance. Wang *et al.* [138] have carried out a systematic study on training targets for speech separation. When using supervised learning, it is critical to choose a suitable training target. In the context of speech separation, a training target should lead to improvement of speech intelligibility and quality, as well as other speech applications. Besides, considering the difficulty of optimization, it should be amenable to training in practice. More efforts are needed to design new training targets for speech separation.

For our binary masking based speech separation, pitch is an important feature for classification, which is estimated using Jin and Wang's algorithm [64]. Because we have developed a pitch tracking algorithm in Chapter 6 and yield better performance than others, it might be useful in improving the results from the classification based speech separation.

Deep neural networks have shown powerful learning capacity in our studies. With multiple hidden layers, they effectively model highly nonlinear representation from raw data. In our experiments, many parameters of neural networks are chosen from development sets, including number of hidden layers, number of hidden units, hidden unit types, and learning rates. However, there is no reason to believe that we have reached the optimal parameters for the neural networks. Although it is a machine learning problem to figure out parameter selection, appropriate choices likely depend on specific applications. Deep learning is still in its infancy. As the progress is being made in this area, deep learning is expected to solve more challenging problems in speech processing.

Speech has rich structure, especially, the temporal dynamics. In this dissertation, we incorporate context information using a window of frames. We have also employed sequence modeling to capture temporal information, such as HMM, RNN, and CRF. In addition, a spectral pattern also encodes speech information, for example, harmonics form prominent structure in the spectral domain. We believe that it is promising to utilize structural information for speech separation and processing, and more work in this direction will likely lead to further improvements in speech separation and processing.

BIBLIOGRAPHY

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Proc. of ECML*, pages 39–50, 2004.

[2] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proc. of ICSLP*, volume 2, pages 1137–1140, 1996.

[3] M. C. Anzalone, L. Calandruccio, K. A. Doherty, and L. H. Carney. Determination of the potential benefit of time-frequency gain manipulation. *Ear and hearing*, 27(5):480–492, 2006.

[4] B. S. Atal. Automatic speaker recognition based on pitch contours. *J. Acoust. Soc. Am.*, 52:1687–1697, 1972.

[5] C. Avendano and H. Hermansky. Study on the dereverberation of speech based on temporal envelope filtering. In *Proc. of ICSLP 1996*, volume 2, pages 889–892. IEEE, 1996.

[6] F. R. Bach and M. I. Jordan. Blind one-microphone speech separation: A spectral learning approach. In *Advances in Neural Information Processing Systems*, volume 17, pages 65–72, 2005.

[7] P. C. Bagshaw, S. M. Hiller, and M. A. Jack. Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching. In *Proc. of Eurospeech*, pages 1003–1006, 1993.

[8] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[9] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of speech corrupted by acoustic noise. In *Proc. of IEEE ICASSP*, volume 4, pages 208–211. IEEE, 1979.

[10] P. Boersma and D. Weenink. PRAAT: Doing Phonetics by Computer (version 4.5). http://www.fon.hum.uva.nl/praat, 2007.

[11] D. C. Boes, F. A. Graybill, and A. M. Mood. *Introduction to the Theory of Statistics, 3rd ed.* McGraw-Hill, New York, NY, USA, 1974.

[12] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust., Speech, Signal Process.*, 27(2):113–120, 1979.

[13] J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic. Training text classifiers with SVM on very few positive examples. Technical Report MSR-TR-2003-34, Microsoft Corp., 2003.

[14] A. S. Bregman. *Auditory scene analysis*, chapter 1. The MIT Press, Cambridge, MA, USA, 1990.

[15] D. S. Brungart, P. S. Chang, B. D. Simpson, and D. L. Wang. Isolating the energetic component of speech-on-speech masking with ideal time-frequency segregation. *J. Acoust. Soc. Am.*, 120(6):4007–4018, 2006.

[16] C. C. Chang and C. J. Lin. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[17] E. C. Cherry. Some experiments on the recognition of speech, with one and with two ears. *J. Acoust. Soc. Am.*, 25(5):975–979, 1953.

[18] W. Chu and A. Alwan. SAFE: a statistical approach to F0 estimation under clean and noisy conditions. *IEEE Trans. Audio, Speech, Language Process.*, 20(3):933–944, 2012.

[19] H. Daumé. Frustratingly easy domain adaptation. In *Proc. of ACL 2007*, volume 45, pages 256–263, 2007.

[20] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proc. of ICML 2007*, pages 209–216, 2007.

[21] A. De Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *J. Acoust. Soc. Am.*, 111:1917, 2002.

[22] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, et al. Recent advances in deep learning for speech research at Microsoft. In *Proc. of IEEE ICASSP*, 2013.

[23] L. Duan, D. Xu, I. W. H. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1667–1680, 2012.

[24] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[25] D. P. W. Ellis. Model-based scene analysis. In D. L. Wang and G. J. Brown, editors, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, chapter 4, pages 115–146. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006.

[26] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[27] Y. Ephraim and D. Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Trans. Acoust., Speech and Signal Process.*, 32(6):1109–1121, 1984.

[28] J. S. Erkelens, R. C. Hendriks, R. Heusdens, and J. Jensen. Minimum mean-square error estimation of discrete Fourier coefficients with generalized gamma priors. *IEEE Trans. Audio, Speech, Lang. Process.*, 15(6):1741–1752, 2007.

[29] G. D. Forney Jr. The Viterbi algorithm. *Proc. of the IEEE*, 61(3):268–278, 1973.

[30] E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar. Conditional random fields in speech, audio, and language processing. *Proceedings of the IEEE*, 101(5):1054–1075, 2013.

[31] Y. Gong. Noise-robust open-set speaker recognition using noise-dependent gaussian mixture classifier. In *Proc. of ICASSP*, volume 1, pages I–133, 2002.

[32] S. Gonzalez and M. Brookes. A pitch estimation filter robust to high levels of noise (PEFAC). In *Proc. EUSIPCO 2011*, 2011.

[33] A. Graves. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012.

[34] D. Griffin and J. S. Lim. Signal estimation from modified short-time fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(2):236–243, 1984.

[35] E. Habets. Room impulse response generator. http://home.tiscali.nl/ehabets/rir_generator.html, 2010.

[36] K. Han and D. L. Wang. An SVM based classification approach to speech separation. In *Proc. of IEEE ICASSP*, pages 5212–5215, 2011.

[37] K. Han and D. L. Wang. A classification based approach to speech segregation. *J. Acoust. Soc. Am.*, 132(5):3475–3483, 2012.

[38] K. Han and D. L. Wang. On generalization of classification based speech separation. In *Proc. IEEE ICASSP*, pages 4541–4544, 2012.

[39] K. Han and D. L. Wang. Learning invariant features for speech separation. In *Proc. of IEEE ICASSP*, pages 7492–7496, 2013.

[40] K. Han and D. L. Wang. Towards generalizing classification based speech separation. *IEEE Trans. Audio, Speech, and Lang. Process.*, 21(1):166–175, 2013.

[41] K. Han, Y. Wang, and D. L. Wang. Learning spectral mapping for speech dereverberation. In *Proc. IEEE ICASSP*, pages 4661–4665, 2014.

[42] K. Han, Y. Wang, and D. L. Wang. Neural networks for supervised pitch tracking in noise. In *Proc. IEEE ICASSP*, pages 1502–1506, 2014.

[43] S. S. Haykin. *Neural networks and learning machines.* Prentice Hall, New York, NY, USA, 2009.

[44] O. Hazrati, J. Lee, and P. C. Loizou. Blind binary masking for reverberation suppression in cochlear implants. *J. Acoust. Soc. Am.*, 133:1607–1614, 2013.

[45] R. C. Hendriks, R. Heusdens, and J. Jensen. MMSE based noise PSD tracking with low complexity. In *Proc. of IEEE ICASSP*, pages 4266–4269, 2010.

[46] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Trans. Speech and Audio Process.*, 2(4):578–589, 1994.

[47] D. J. Hermes. Measurement of pitch by subharmonic summation. *J. Acoust. Soc. Am.*, 83:257, 1988.

[48] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.*, 2012.

[49] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[50] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[51] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[52] G. Hu. 100 nonspeech sounds, 2006. http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html.

[53] G. Hu. *Monaural speech organization and segregation.* Ph.D. dissertation, The Ohio State University, Columbus, OH, 2006.

[54] G. Hu and D. L. Wang. Monaural speech segregation based on pitch tracking and amplitude modulation. *IEEE Trans. Neural Netw.*, 15(5):1135–1150, 2004.

[55] G. Hu and D. L. Wang. Auditory segmentation based on onset and offset analysis. *IEEE Trans. Audio, Speech, Lang. Process.*, 15(2):396–405, 2007.

[56] G. Hu and D. L. Wang. Segregation of unvoiced speech from nonspeech interference. *J. Acoust. Soc. Am.*, 124:1306–1319, 2008.

[57] G. Hu and D. L. Wang. A tandem algorithm for pitch estimation and voiced speech segregation. *IEEE Trans. Audio, Speech, Lang. Process.*, 18(8):2067–2079, 2010.

[58] K. Hu and D. L. Wang. Unvoiced speech segregation from nonspeech interference via CASA and spectral subtraction. *IEEE Trans. Audio, Speech, Lang. Process.*, 19(6):1600–1609, 2011.

[59] K. Hu and D. L. Wang. SVM-based separation of unvoiced-voiced speech in cochannel conditions. In *Proc. of ICASSP*, pages 4545–4548. IEEE, 2012.

[60] F. Huang and T. Lee. Pitch estimation in noisy speech using accumulated peak spectrum and sparse estimation technique. *IEEE Trans. Speech, Audio Process.*, 21(3):99–109, 2013.

[61] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon. Metric and kernel learning using a linear transformation. *J. Mach. Learn. Res.*, 13:519–547, 2012.

[62] J. Jensen and R. C. Hendriks. Spectral magnitude minimum mean-square error estimation using binary and continuous gain functions. *IEEE Trans. Audio, Speech, and Lang. Process.*, 20(1):92–102, 2012.

[63] Z. Jin and D. L. Wang. A supervised learning approach to monaural segregation of reverberant speech. *IEEE Trans. Audio, Speech, Lang. Process.*, 17(4):625–638, 2009.

[64] Z. Jin and D. L. Wang. A multipitch tracking algorithms for noisy and reverberant speech. In *Proc. of IEEE ICASSP*, pages 4218–4221, 2010.

[65] Z. Jin and D. L. Wang. HMM-based multipitch tracking for noisy and reverberant speech. *IEEE Trans. Audio, Speech, Lang. Process.*, 19(5):1091–1102, 2011.

[66] Z. Jin and D. L. Wang. Reverberant pitch evaluation corpus. http://www.cse.ohio-state.edu/pnl/shareware/jin1-taslp11, 2011.

[67] V. Kalatzis and C. Petit. The fundamental and medical impacts of recent progress in research on hereditary hearing loss. *Human molecular genetics*, 7(10):1589–1597, 1998.

[68] S. Kamath and P. Loizou. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In *Proc. of IEEE ICASSP*, volume 4, pages 4164–4164, 2002.

[69] G. Kim and P. C. Loizou. Improving speech intelligibility in noise using environment-optimized algorithms. *IEEE Trans. Audio, Speech, Lang. Process.*, 18(8):2080–2090, 2010.

[70] G. Kim, Y. Lu, Y. Hu, and P. C. Loizou. An algorithm that improves speech intelligibility in noise for normal-hearing listeners. *J. Acoust. Soc. Am.*, 126:1486–1494, 2009.

[71] B. Kingsbury. Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In *Proc. of ICASSP*, pages 3761–3764, 2009.

[72] B. Kingsbury and N. Morgan. Recognizing reverberant speech with RASTA-PLP. In *Proc. of IEEE ICASSP 1997*, volume 2, pages 1259–1262. IEEE, 1997.

[73] K. Kokkinakis, O. Hazrati, and P. C. Loizou. A channel-selection criterion for suppressing reverberation in cochlear implants. *J. Acoust. Soc. Am.*, 129:3221–3232, 2011.

[74] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proc. of IEEE CVPR 2011*, pages 1785–1792, 2011.

[75] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the ICML*, pages 282–289, 2001.

[76] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama. Fast signal reconstruction from magnitude stft spectrogram based on spectrogram consistency. In *Proc. of International Conference on Digital Audio Effects DAFx*, volume 10, 2010.

[77] B. S. Lee and D. P. W. Ellis. Noise robust pitch tracking by subband autocorrelation classification. In *Proc. of Interspeech*, 2012.

[78] D. D. Lee and S. H. S. Algorithms for nonnegative matrix factorization. volume 13, pages 556–562, 2001.

[79] J. Li, D. Yu, J.-T. Huang, and Y. Gong. Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM. In *Proc. of SLT*, 2012.

[80] N. Li and P. C. Loizou. Factors influencing intelligibility of ideal binary-masked speech: Implications for noise reduction. *J. Acoust. Soc. Am.*, 123(3):1673–1682, 2008.

[81] R. P. Lippmann. Speech recognition by machines and humans. *Speech communication*, 22(1):1–15, 1997.

[82] P. Lockwood and J. Boudy. Experiments with a Nonlinear Spectral Subtractor (NSS), Hidden Markov Models and the projection, for robust speech recognition in cars. *Speech Communication*, 11(2-3):215–228, 1992.

[83] P. C. Loizou. *Speech Enhancement: Theory and Practice.* Taylor & Francis, New York, NY, USA, 2007.

[84] J. Ma, Y. Hu, and P. C. Loizou. Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions. *J. Acoust. Soc. Am.*, 125(5):3387–3405, 2009.

[85] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Ng. Recurrent neural networks for noise reduction in robust ASR. In *Proc. of Interspeech*, 2012.

[86] C. McGonegal, L. Rabiner, and A. Rosenberg. A semiautomatic pitch detector (SAPD). *IEEE Trans. Acoust., Speech, Signal Process.*, 23(6):570–574, 1975.

[87] M. Miyoshi and Y. Kaneda. Inverse filtering of room acoustics. *IEEE Trans. Audio, Speech, and Lang. Process.*, 36(2):145–152, 1988.

[88] A. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Trans. Audio, Speech, Lang. Process.*, 20(1):14–22, 2012.

[89] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[90] T. Nakatani, S. Amano, T. Irino, K. Ishizuka, and T. Kondo. A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments. *Speech Communication*, 50(3):203–214, 2008.

[91] A. Narayanan and D. Wang. Joint noise adaptive training for robust automatic speech recognition. In *Proc. ICASSP, ~~to appear~~*, pages 2523–2527. IEEE, 2014.

[92] A. Narayanan and D. L. Wang. A CASA-based system for long-term SNR estimation. *IEEE Trans. Audio, Speech, and Lang. Process.*, 20(9):2518–2527, 2012.

[93] P. Naylor and N. Gaubitch, editors. *Speech dereverberation.* Springer, 2010.

[94] S. T. Neely and J. B. Allen. Invertibility of a room impulse response. *J. Acoust. Soc. Am.*, 66:165, 1979.

[95] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-time signal processing (2nd ed.).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1999.

[96] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval. Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *IEEE Trans. Audio, Speech, and Lang. Process.*, 15(5):1564–1578, 2007.

[97] A. Ozerov, E. Vincent, and F. Bimbot. A general flexible framework for the handling of prior information in audio source separation. *IEEE Trans. Audio, Speech, Lang. Process.*, 20(4):1118–1133, 2012.

[98] K. Paliwal and A. Basu. A speech enhancement method based on kalman filtering. In *Proc. of IEEE ICASSP*, volume 12, pages 177–180, 1987.

[99] R. D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice. An efficient auditory filterbank based on the gammatone function. Technical Report No. 2341, MRC Applied Psychology Unit, 1988.

[100] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pages 61–74. The MIT Press, Cambridge, MA, USA, 1999.

[101] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The Kaldi speech recognition toolkit. In *Proc. ASRU*, pages 1–4, 2011.

[102] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset shift in machine learning.* The MIT Press, 2009.

[103] M. H. Radfar and R. M. Dansereau. Single-channel speech separation using soft mask filtering. *IEEE Trans. Audio, Speech, and Lang. Process.*, 15(8):2299–2310, 2007.

[104] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proc. of ICML 2007*, pages 759–766, 2007.

[105] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *Proc. of ICASSP*, volume 2, pages 749–752. IEEE, 2001.

[106] N. Roman and D. L. Wang. Pitch-based monaural segregation of reverberant speech. *J. Acoust. Soc. Am.*, 120:458, 2006.

[107] N. Roman, D. L. Wang, and G. J. Brown. Speech segregation based on sound localization. *J. Acoust. Soc. Am.*, 114(4):2236–2252, 2003.

[108] N. Roman and J. Woodruff. Intelligibility of reverberant noisy speech with ideal binary masking. *J. Acoust. Soc. Am.*, 130:2153, 2011.

[109] N. Roman and J. Woodruff. Speech intelligibility in reverberation with ideal binary masking: Effects of early reflections and signal-to-noise ratio threshold. *J. Acoust. Soc. Am.*, 133(3):1707–1717, 2013.

[110] E. H. Rothauser, W. D. Chapman, N. Guttman, K. S. Nordby, H. R. Silbiger, G. E. Urbanek, and M. Weinstock. IEEE recommended practice for speech quality measurements. *IEEE Trans. Audio Electroacoustics*, 17:227–246, 1969.

[111] S. T. Roweis. One microphone source separation. In *Advances in Neural Information Processing Systems*, pages 793–799, 2001.

[112] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[113] S. O. Sadjadi and J. H. L. Hansen. Hilbert envelope based features for robust speaker identification under reverberant mismatched conditions. In *Proc. of ICASSP 2011*, pages 5448–5451. IEEE, 2011.

[114] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. of ECCV 2010*, pages 213–226, 2010.

[115] H. Sameti, H. Sheikhzadeh, L. Deng, and R. L. Brennan. HMM-based strategies for enhancement of speech signals embedded in nonstationary noise. *IEEE Trans. Speech and Audio Process.*, 6(5):445–455, 1998.

[116] P. Scalart and J. Filho. Speech enhancement based on a priori signal to noise estimation. In *Proc. of IEEE ICASSP*, volume 2, pages 629–632, 1996.

[117] M. R. Schroeder. Period histogram and product spectrum: New methods for fundamental-frequency measurement. *J. Acoust. Soc. Am.*, 43:829, 1968.

[118] M. L. Seltzer, B. Raj, and R. M. Stern. A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition. *Speech Communication*, 43(4):379–393, 2004.

[119] J. Sohn, N. S. Kim, and W. Sung. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3, 1999.

[120] A. Sun, E. P. Lim, and Y. Liu. On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, 48(1):191–201, 2009.

[121] I. Sutskever. *Training Recurrent Neural Networks.* Ph.D. dissertation, University of Toronto, Toronto, Canada, 2013.

[122] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Trans. Audio, Speech, Lang. Process.*, 19(7):2125–2136, 2011.

[123] Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey. Discriminative methods for noise robust speech recognition: A CHiME challenge benchmark. *Proc. CHiME-2013*, pages 19–24, 2013.

[124] D. Talkin. A robust algorithm for pitch tracking (RAPT). *Speech coding and synthesis*, 495:518, 1995.

[125] J. Tchorz and B. Kollmeier. Estimation of the signal-to-noise ratio with amplitude modulation spectrograms. *Speech Commun.*, 38(1-2):1–17, 2002.

[126] J. Tchorz and B. Kollmeier. SNR estimation based on amplitude modulation analysis with applications to noise suppression. *IEEE Trans. Speech Audio Process.*, 11(3):184–192, 2003.

[127] V. N. Vapnik. *The nature of statistical learning theory.* Springer, Inc., New York, NY, USA, 2000.

[128] A. Varga and H. J. M. Steeneken. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12(3):247–251, 1993.

[129] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni. The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines. In *Proc. of ICASSP 2013*, pages 126–130, 2013.

[130] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Machine Learning Res.*, 11:3371–3408, 2010.

[131] O. Vinyals, S. V. Ravuri, and D. Povey. Revisiting recurrent neural networks for robust ASR. In *Proc. of ICASSP*, pages 4085–4088, 2012.

[132] T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. Audio, Speech, and Lang. Process.*, 15(3):1066–1074, 2007.

[133] D. L. Wang. On ideal binary mask as the computational goal of auditory scene analysis. In P. Divenyi, editor, *Speech separation by humans and machines*, pages 181–197. Kluwer Academic Pub., 2005.

[134] D. L. Wang and G. J. Brown, editors. *Computational auditory scene analysis: Principles, algorithms and applications*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006.

[135] D. L. Wang and G. J. Brown. Fundamentals of Computational Auditory Scene Analysis. In D. L. Wang and G. J. Brown, editors, *Computational auditory scene analysis: Principles, algorithms and applications*, chapter 1, pages 1–37. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006.

[136] D. L. Wang, U. Kjems, M. S. Pedersen, J. B. Boldt, and T. Lunner. Speech intelligibility in background noise with ideal binary time-frequency masking. *J. Acoust. Soc. Am.*, 125:2336–2347, 2009.

[137] Y. Wang, K. Han, and D. L. Wang. Exploring monaural features for classification-based speech segregation. *IEEE Trans. Audio, Speech, Lang. Process.*, 21(2):270–279, 2012.

[138] Y. Wang, A. Narayanan, and D. Wang. On training targets for supervised speech separation. Technical report, Technical Report OSU-CISRC-2/14-TR05, Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, USA, 2014. Available: ftp://ftp. cse. ohio-state. edu/pub/techreport/2014/TR05. pdf. 11, 26, 27, 28.

[139] Y. Wang and D. L. Wang. Feature denoising for speech separation in unknown noisy environments. In *Proc. of IEEE ICASSP 2013*, pages 7472–7476. IEEE, 2013.

[140] Y. Wang and D. L. Wang. Towards scaling up classification-based speech separation. *IEEE Trans. Audio, Speech, Lang. Process.*, 21(7):1381–1390, 2013.

[141] R. J. Weiss and D. P. W. Ellis. Estimating single-channel source separation masks: Relevance vector machine classifiers vs. pitch-based masking. In *Proc. Workshop on Statistical and Perceptual Audition*, pages 31–36, 2006.

[142] R. J. Williams and J. Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990.

[143] G. Wu and E. Y. Chang. KBA: kernel boundary alignment considering imbalanced data distribution. *IEEE Trans. Knowledge Data Eng.*, 17(6):786–795, 2005.

[144] K. Wu and D. G. Childers. Gender recognition from speech. part i: Coarse analysis. *J. Acoust. Soc. Am.*, 90:1828–1840, 1991.

[145] M. Wu and D. L. Wang. A one-microphone algorithm for reverberant speech enhancement. In *Proc. of IEEE ICASSP 2003*, pages 844–847. IEEE, 2003.

[146] M. Wu and D. L. Wang. A two-stage algorithm for one-microphone reverberant speech enhancement. *IEEE Trans. Audio, Speech, and Lang. Process.*, 14(3):774–784, 2006.

[147] M. Wu, D. L. Wang, and G. J. Brown. A multipitch tracking algorithm for noisy speech. *IEEE Trans. Speech, Audio Process.*, 11(3):229–241, 2003.

[148] D. Y. Zhao and W. B. Kleijn. HMM-based gain modeling for enhancement of speech in noise. *IEEE Trans. Audio, Speech, and Lang. Process.*, 15(3):882–892, 2007.

[149] X. Zhao, Y. Shao, and D. L. Wang. CASA-based robust speaker identification. *IEEE Trans. Audio, Speech, Lang. Process.*, 20(5):1608–1616, 2012.

[150] X. Zhao, Y. Wang, and D. L. Wang. Robust speaker identification in noisy and reverberant conditions. *IEEE Trans. Audio, Speech, Lang. Process.*, 22:836–845, 2014.

[151] V. Zue, S. Seneff, and J. Glass. Speech database development at MIT: TIMIT and beyond. *Speech Communication*, 9(4):351–356, 1990.