ROBUST ACOUSTIC MODELING AND FRONT-END DESIGN FOR DISTANT SPEECH RECOGNITION

by

Seyedmahdad Mirsamadi

APPROVED BY SUPERVISORY COMMITTEE:

John H. L. Hansen, Chair

P. K. Rajasekaran

Carlos Busso

Yang Liu

Chin-Tuan Tan

Dedicated to my parents, Mansour and Soheila

ROBUST ACOUSTIC MODELING AND FRONT-END DESIGN FOR DISTANT SPEECH RECOGNITION

by

SEYEDMAHDAD MIRSAMADI, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2017

ACKNOWLEDGMENTS

First and foremost, I would like to express my special appreciation and thanks to my PhD advisor, Dr. John H. L. Hansen, for his continuous support and remarkable patience, his valuable guidance, and his unique professional and moral standards in directing academic research. The opportunity of working with him in the past few years has been an extremely valuable experience which not only made me a far better researcher in speech technology, but also helped me learn many other aspects of being an effective researcher and contributor by observing his unique style of leadership and management.

I would like to express my sincere gratitude to my committee members, Dr. P. K. Rajasekaran, Dr. Carlos Busso, Dr. Yang Liu and Dr. Chin-Tuan Tan for their precious time and invaluable comments on my research.

It was a pleasure working with my fellow researchers and colleagues at the Center for Robust Speech Systems (CRSS). Working together with them was an important aspect of my PhD experience. In particular, I would like to thank Navid Shokouhi, Abhinav Misra, Chengzhu Yu, Qian Zhang, and Shabnam Ghaffarzadegan for all the fruitful scentific discussions, support, and friendship throughout my years at CRSS. I am also thankful to CRSS alumni Dr. Omid Sadjadi, Dr. Oldooz Hazrati, and Dr. Ali Ziaei for all their valuable help during my first years as a PhD student at CRSS.

This acknowledgement would be incomplete without thanks to my mentors at Microsoft Research (MSR) during my summer internships, Dr. Ivan Tashev and Dr. Cha Zhang. The opportunity of working with them at MSR greatly enriched my PhD experience and helped me become a better researcher.

I am also appreciative of the efforts of the administrative staff at the Electrical Engineering Department at UT-Dallas. My special thanks goes to Ms. Tammy Emery for her continuous help at the different stages of my studies at CRSS.

Last but not least, I would like to thank my parents, Mansour and Soheila, for their continuous support throughout my life, particularly during the years of studying abroad. I am very grateful for their significant role in enabling me to initiate my studies at UT-Dallas, and also for their invaluable support during my PhD years.

July 2017

ROBUST ACOUSTIC MODELING AND FRONT-END DESIGN FOR DISTANT SPEECH RECOGNITION

Seyedmahdad Mirsamadi, PhD
The University of Texas at Dallas, 2017

Supervising Professor: John H. L. Hansen, Chair

In recent years, there has been a significant increase in the popularity of voice-enabled technologies which use human speech as the primary interface with machines. Recent advancements in acoustic modeling and feature design have increased the accuracy of Automatic Speech Recognition (ASR) to levels that enable voice interfaces to be used in many applications. However, much of the current performance is dependent on the use of close-talking microphones, (i.e., scenarios in which the user speaks directly into a hand-held or body-worn microphone). There is still a rather large performance gap experienced in distant-talking scenarios in which speech is recorded by far-field microphones that are placed at a distance from the speaker. In such scenarios, the distorting effects of distance (such as room reverberation and environment noise) make the recognition task significantly more challenging. In this dissertation, we propose novel approaches for designing a distant-talking ASR front-end as well as training robust acoustic models to reduce the existing gap between far-field and close-talking ASR performance. Specifically, we i) propose a novel multi-channel front-end enhancement algorithm for improved ASR in reverberant rooms using distributed non-uniform microphone arrays with random unknown locations; ii) propose a novel neural network model training approach using adversarial training to improve the robustness of multi-condition acoustic models that are trained directly on far-field data; iii) study alter-

nate neural network adaptation strategies for far-field adaptation to the acoustic properties of specific target environments. Experimental results are provided based on far-field benchmark tasks and datasets which demonstrate the effectiveness of the proposed approaches for increasing far-field robustness in ASR. Based on experiments using reverberated TIMIT sentences, the proposed multi-channel front-end provides WER improvements of +21.5% and +37.7% in two-channel and four-channel scenarios over a single-channel scenario in which the channel with best signal quality is selected. On the acoustic modeling side and based on results of experiments on AMI corpus, the proposed multi-domain training approach provides a relative charachter error rate reduction of +3.3% with respect to a conventional multi-condition trained baseline, and +25.4% with respect to a clean-trained baseline.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AIR	Aachen Impulse Response
AMI	Augmented Multi-party Interaction
ASR	Automatic Speech Recognition
BLSTM	Bi-directional Long Short Term Memory
CER	Character Error Rate
CMVN	Cepstral Mean and Variance Normalization
CNTF	Convolutive Nonnegative Tensor Factorization
CTC	Connectionist Temporal Classification
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
DSP	Digital Signal Processing
DRR	Direct to Reverberant Ratio
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
IHM	Independent Headset Microphone
KLD	Kullback Leibler Divergence
LDA	Linear Discriminant Analysis
LDC	Linguistic Data Consortium
LM	Language Model
LSTM	Long Short Term Memory

LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A Posteriori
MFB	Mel Filter Bank
MFCC	Mel Frequency Cepstral Coefficients
MTL	Multi Task Learning
NMF	Nonnegative Matrix Factorization
PCA	Principal Component Analysis
PLP	Perceptual Linear Prediction
RIR	Room Impulse Response
RM	Resource Management
RNN	Recurrent Neural Network
SDM	Single Distant Microphone
SGD	Stochastic Gradient Descent
SNR	Signal to Noise Ratio
STFT	Short-Time Fourier Transform
SWB	Switchboard
WER	Word Error Rate
WFST	Weighted Finite State Transducer
WSJ	Wall Street Journal

CHAPTER 1

INTRODUCTION

The increasing popularity of voice-enabled devices and digital assistants in recent years has caused Automatic Speech Recognition (ASR) technology to leave the research labs and play a central role as the dominant human-machine interface. Combined with Natural Language Processing (NLP), speech recognition is reshaping the way we interact with machines. Today, almost all our mobile devices (phones, tablets, laptops, etc.) have one or more built-in microphones. With ongoing advancements in the development of Internet of Things (IoT), more and more elements will be added to this network of voice-enabled devices. As a result, many technological companies are redefining their services and products to use speech recognition (and synthesis) as the primary way of interacting with the user. The accuracy of speech recognition is a key factor in the usability of such systems, because the operation of the rest of the dialog system is critically dependentent on the recognized content.

Research on speech recognition has a history of a few decades [2]. It started in 1950s with limited-vocabulary systems aimed at recognizing a small set of words in rather ideal acoustic conditions. Since then, decades of research in both Digital Signal Processing (DSP) and Machine Learning (ML) has brought about a drastic improvement in ASR performance. The recognition accuracy has continuosly improved, with periods of slow gradual improvements as well as occasional leaps in performance resulting from breakthroughs in modeling or representation techniques. As the performance improves, the test conditions and application scopes are constantly redefined to include more realistic conditions.

Figure 1.1 shows a summary of the improvements in Word Error Rate (WER) on a few popular speech corpora. These datasets were each designed to address a new problem and take the application scenario into a more realistic level. Focus has shifted from small-vocabulary systems to Large Vocabulary Continuous Speech Recognition (LVCSR), from read speech to conversational speech, and from clean close-talking speech to noisy and

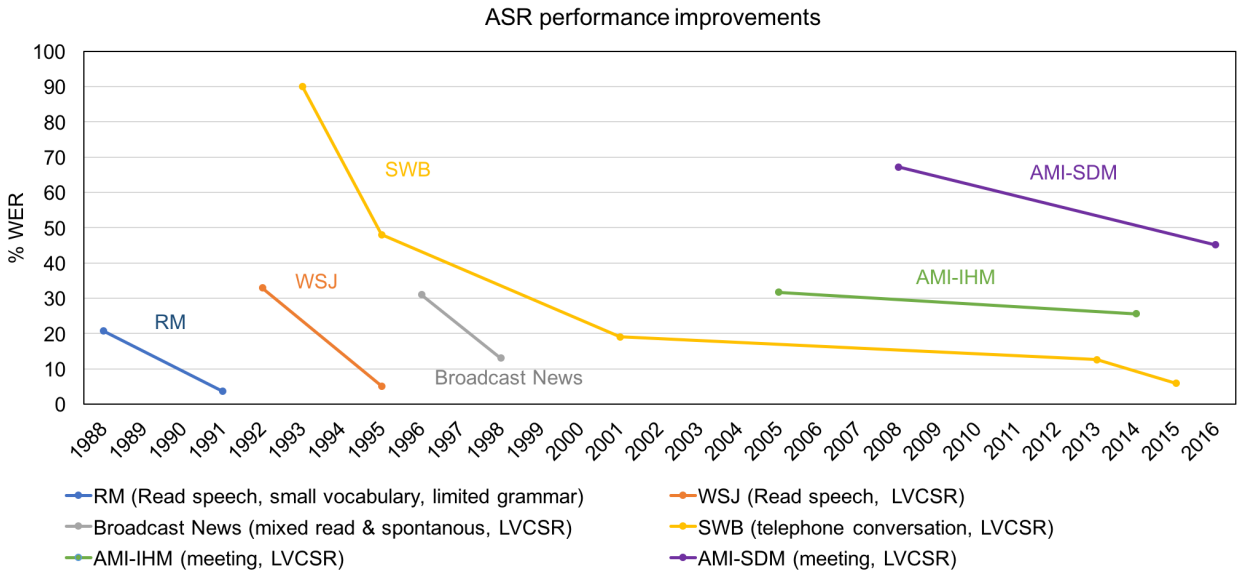


Figure 1.1. Accuracy improvement of ASR systems on increasingly challenging and realistic scenarios. Resource Management (RM) is a small-vocabulary read-speech dataset with limited lexicon and fixed grammar [3]. Wall Street Journal (WSJ) read-speech corpus [4] is one of the first corpora aimed at benchmarking LVCSR. Broadcast News is a collection of radio and television material speech (mixed read and spontaneous). Switchboard (SWB) [5] is a corpus of conversational telephone speech. The Augmented Multi-party Interaction (AMI) corpus [6] is a dataset of conversational speech in meetings recorded both using headset (IHM) and distant (SDM) microphones. This illustration is based on data from [7–10]

reverberant speech. The adoption of Hidden Markov Models (HMM) for statistical modeling of speech significantly imporved ASR accuracy on read speech. However, recognizing conversational speech in natural interactions (e.g., the Switchboard corpus [5]) remained a challenge. The nest breakthrough was the adoption of Deep Neural Network (DNN) acoustic models which were found capable of vastly outperforming conventional Gaussian Mixture Models (GMM) [11]. Further advancements in DNN based acoustic modeling such as the use of convolutional and recurrent layers, improved optimization algorithms and i-vector based speaker adaptive training [12], have now made it possible to achieve significantly improved accuracy on conversational speech. Error rates as low as 5.5% have recently been reported on the Switchboard corpus [9, 13], which is very close to human performance on transcribing convesational speech.

Given the improvements provided by DNN-based acoustic models, the availability of large speech datasets, and the possibility of training large neural networks on such datasets using efficient hardware such as Graphical Processing Units (GPU), the problem of close-talking single-speaker ASR in reasonably matched acoustic conditions is largely considered to be solved today. However, there are still significant hurdles in widespread deployment of speech recognition technology in day-to-day lives, because it requires relaxing many of the constraints assumed for the task. Among the remaining challenges, the problem of distant (far-field) speech recognition is one of the most significant hurdles which currently limits practical applications of ASR. The goal of this dissertation is to provide solutions for reducing the gap between distant-talking and close-talking ASR performance.

1.1 Motivation

Despite the advancements in acoustic modeling which brough the recent improvements in ASR accuracy, it is still necessary to use close-talking microphones (headset, lapel, or hand-held microphones) to achieve satisfactory performance. In situations where a distant (far-field) microphone is to be used, the distortions in the received signal lead to considerable performance degradation. Typical WERs on distant microphones are twice as high as the close-talking counterparts. For conversational speech and even with state-of-the-art DNN-based acoustic models, the error rates on distant microphones are usually high enough to render the ASR system unusable. This is a major limitation because in many applications, wearing a headset microphone or speaking directly into a microphone at a fixed position is impossible or too limiting. It is desirable to let users talk to machines while freely moving within the environment, without having to wear cumbersome recording or transmitting devices. There are numerous applications that can enormously benefit from the availability of an accurate distant-talking ASR, including

- Automatic annotation of user-generated videos (e.g. YouTube videos), often recorded from a distance using a mobile device.
- Meeting transcription which enables archiving and searching of meeting contents.
- Remotely operating robots or other consumer products (TV, mobile devices, etc.)
- Voice-operated technologies in cars (e.g., sound or navigation systems), which let the driver focus on the road while operating these devices at the same time.

As a result, the focus in research is now shifting to distant-talking (or far-field) ASR, which can potentially bring in a whole new set of applications. Figure 1.1 shows the accuracies obtained from both close-talking and distant microphones in a meeting transcription task on the AMI corpus [6] (which has become a fairly standard and popular dataset for distant ASR). The best results reported on the Single Distant Microphone (SDM) task on this dataset have a WER of around 45% [14, 15], which shows the significant challenge that far-field conversational speech still poses to today's ASR systems.

There are various factors leading to degraded performance in distant speech recognition:

- **Room reverberation** is by far the most serious challenge in distant ASR. The reflections of the sound wave from the walls or other surfaces in an enclosed environment causes the microphone to capture multiple attenuated replicas of the original speech signal with different time delays and power levels. These replicas serve as non-sationary and non-Gaussian noise which is correlated with the direct-path speech component.
- **Environment noise** leads to lower Signal to Noise Ratio (SNR) on distant microphones, becuase the intensity of the captured sound is inversely proportional to the distance it travels (while the level of ambient noise remains constant regardless of microphone location, and the level of non-stationary interferences depend on the relative location of the noise source and the microphone).

- **Simultanous speech from multiple speakers** is another important challenge in ASR. Again, this is usually not an issue with close-talking microphones because the energy of the primary speaker is dominant on the microphone. Despite the currently active research on co-channel speech [16, 17] and blind source separation [18], speech recognition in the presence of competing speakers remains a largely unsolved problem today.

- **Speaker's head orientation** can introduce additional channel distortions in reverberant rooms. The human speech is a directional sound source which emits more energy in the forward direction. When a speaker is facing a direction other than the microphone's location, the direct path component is further attenuated relative to the reverberation components.

- **Speaker movements** can introduce time-varying components to the acoustic transfer function between the speaker and the microphone. The transfer function can significantly change even with small movements of the speaker. Similarly, the microphone's location may change (e.g., in the case of a moving robot), which will create a similar time-varying effect.

- **The perception of distance** can also influence the way humans talk. Two different application scenarios can be distinguished in far-field ASR. If the goal is to recognize a conversation between humans (e.g., in meeting transcription), the uttered speech characteristics are independent of the microphone distance. However, in applications where a human user is talking directly to a distant machine (e.g., communicating with robots), users often try to raise their voice if they are aware of the distance. This will change the speech production characteristics, resulting in further mismatch with an acoustic model that is trained on neutral speech.

To achieve robust distant speech recognition, solutions are needed to address the above problems in the different stages of recognition. On the feature extraction front-end, multi-channel or single-channel feature enhancement techniques are needed to compensate for the non-stationary and context-dependent distortion caused by reverberation. On the acoustic model side, modeling techniques are needed which present better robustness to the types of distortion in far-field speech. Finally, adaptation mechanisms are needed to tailor general-purpose acoustic models to specific rooms and environment properties.

1.2 Dissertation Contributions

In this dissertation, we aim to reduce the gap between close-talking and distant-talking ASR performance by providing multiple solutions to improve far-field robustness at different stages of the recognition pipeline. The main proposed solutions are as follows.

- **A multichannel front-end for effective integration of acoustic information from multiple independent recording devices (CNTF algorithm):** A major area of research to achieve robustness in far-field ASR is multichannel front-end processing to obtain estimates of the clean speech features. The majority of existing research in this area focuses on uniform microphone arrays (i.e. closely spaced microphones that are designed and calibrated in advance for beamforming, localization, noise cancellation, etc.). In contrast, our focus will be on independent recording devices that can be distributed across a room, relaxing the constraint of having compact unifrom arrays. The signals recorded by these distributed microphones do not have any synchrony or meaningful phase information among them, and thus they require fundamentally different approaches to combine their signals into a single set of features for ASR. We propose a mutichannel dereverberation method based on Convolutive Nonnegative Tensor Factorization (CNTF) suitable for such distributed scenarios.

- **Multi-domain adversarial training of neural network acoustic models for increased robustness in far-field ASR:** If a large corpus of far-field speech recordings from different rooms with different acoustic properties is available (referred to as multi-domain or multi-condition data), it is possible to train DNN acoustic models directly on such data without any front-end processing. Most existing studies in this area compile data from all different recording environments into a single train set, ignoring the environment labels during training. We propose a novel training strategy for end-to-end RNN acoustic models which uses the available meta-data regarding the recording environment of each utterance in order to obtain increased robustness. This is achieved by enforcing increased invariance between the different environments (domains) by tuning a subset of network parameters adversarially with respect to a domain classifier.

- **Deep Neural Network (DNN) acoustic model adaptation for increased robustness in specific target environments:** Model adaptation is very useful for improving the performance of acoustic models for specific target environments. However, most existing studies on adaptation focus on conventional GMM-HMM models, and also on the task of speaker adaptation. We provide a thorough study on possible DNN adaptation startegies for far-field ASR. Focusing on the linear transformation approach (which provides best results in practice), we demonstrate how to choose specific parameter subsets in a network which result in best adaptation performance.

1.3 Dissertation Outline

The rest of this dissertation is organized as follows.

- **Chapter 2** provides a brief review on state-of-the-art ASR and its different components including feature extraction, acoustic modeling, language modeling and decoding. It highlights the differences between traditional HMM-based acoustic models and

the more recent RNN-based end-to-end models. Moreover, we provide some relevant background specifically for the task of distant speech recognition, including the far-field speech model, reasons why conventional robustness techniques are not useful for far-field distortion, and existing approaches to address the far-field problem.

- **Chapter 3** presents CNTF algorithm, which is our proposed multichannel solution for improving far-field ASR through integration of acoustic information from multiple independent microphones. It explains the motivations of using distributed recording devices instead of unifrom microphone arrays and the challenges involved. It then presents the tensor model for the time-frequency representations of reverberant speech, and formulates a dereverberation algorithm based on Convolutive Nonnegative Tensor Factorization. Finally, results of speech recognition experiments are provided which show the effectiveness of the proposed approach in highly reverberant environments.

- **Chapter 4** presents our proposed RNN training strategy to improve environmental robustness with multi-domain training data. We propose a novel training method for RNN-CTC models based on tuning a subset of RNN parameters adversarially with respect to a domain classifier built on top of hidden features. The proposed approach can use available information about the recording environment of each utterance in order to enforce increased domain invariance in the RNN hidden layers. We provide results on a meeting trasncription task which demonstrate the effectiveness of the proposed approach.

- **Chapter 5** provides adaptation strategies for DNN-based acoustic models to improve performance in specific target environments. Most existing adaptation strategies in the literature are specifically designed for GMM-HMMs and cannot be applied to DNN-based models. Moreover, they are usually designed for the task of speaker adaptation and do not consider the types of distortion in far-field ASR. We present different

adaptation approaches for DNN-based acoustic models that are appropriate for the task of environment adaptation. We peovide recognition results both with DNN-HMM hybrids and with end-to-end RNNs which show the effectiveness of far-field adaptation for both models.

- **Chapter 6** concludes this dissertation and provides a summary of the proposed contributions. It also discusses some open research problems and provides future research directions.

CHAPTER 2

BACKGROUND

This chapter provides a brief overview of automatic speech recognition, and reviews the role and functionality of its different components. Design choices and recent advancements in different components are briefly discussed, with a focus on recent developments in deep learning research which has fundamentally changed the speech recognition pipeline. Note that this is by no means a comprehensive review on ASR technology. The intention here is to familiarize the reader with those concepts that are directly related to far-field robustness, and which are needed in subsequent chapters in order to formulate the proposed solutions. For more details on ASR basics, the reader is encouraged to refer to [19–21].

2.1 Speech recognition fundamentals

The goal in speech recognition is to map a sequence of acoustic observations $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$ to a corresponding sequence of words $\mathbf{w} = [w_1, w_2, \cdots, w_N]$ which maximize the posterior probability

$$p(\mathbf{w}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{x})}. \tag{2.1}$$

The denominator in Eq. (2.1) is a constant term based only on the fixed acoustic observation, and thus our task in ASR is to find the sequence of words which maximize $p(\mathbf{x}|\mathbf{w})p(\mathbf{w})$. The term $p(\mathbf{w})$ encodes our knowledege on which sequences of words are more probable than others (independent of the acoustic observation), which is referred to as the Language Model (LM). The term $p(\mathbf{x}|\mathbf{w})$ is the likelihood of the observed acoustic features assuming a certain sequence of words $\mathbf{w}$. This is determined by the acoustic model (AM), which specifies how each word (or sub-word unit) is represented in the feature space. If sub-word units are used (which is necessary in LVCSR), a separate component is required to determine the sequence of sub-word units for each word (referred to as the lexicon or dictionary). Throughout this

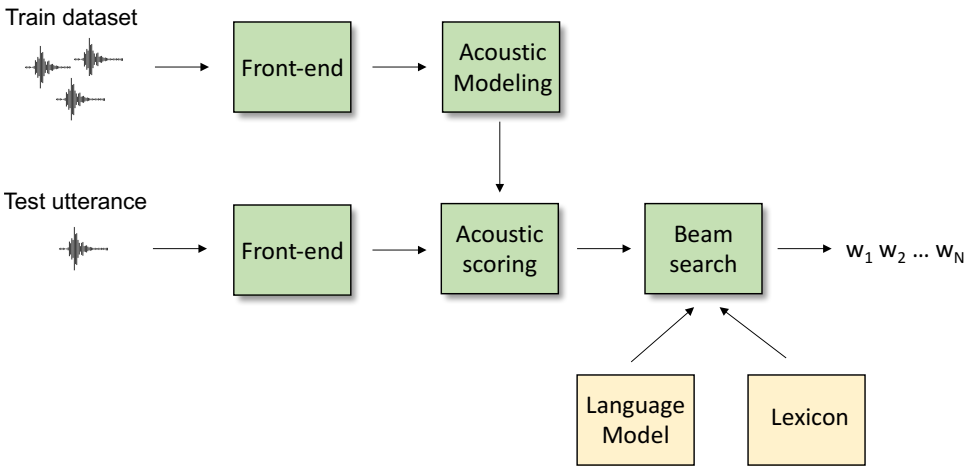


Figure 2.1. An overview of the different components in ASR

dissertation, we collectively refer to these sub-word units as *labels*. For HMM-based acoustic models, these labels are often context-dependent HMM states, better known as *senones*. For RNN-based acoustic models, the labels can be monophones or text characters.

Figure 2.1 shows a block diagram of a typical speech recognition system. The system is developed using a training database of transcribed utterances which is supposed to cover the diversity of the intended test scenario as much as possible (in terms of speaker, gender, age, environment, etc.). The ASR front-end is responsible for converting speech waveforms into a sequence of feature vectors. These features are designed to encode phoneme information while rejecting other sources of variability. Some popular choices of features are Mel filter-bank (MFB) features, Mel Frequency Cepstral Coefficients (MFCC), and Perceptual Linear Prediction (PLP) coefficients. The front-end can optionally include enhancement algorithms to compensate for different types of distortion such as reverberation, noise, etc. The resulting feature vectors are used to construct models for each word or sub-word unit. For large vocabulary tasks, it is not feasible to use word models. Therefore, a lexicon is needed to map each word in the transcripts to a corresponding sequence of sub-word units. Models are built for these sub-word units which can be phonemes, context-dependent phones (e.g., triphones), or more recently, text characters corresponding to each word. There are three

possible choices for the acoustic model architecture, namely GMM-HMM, DNN-HMM, and RNN models. The task of these models is to provide the likelihood of an acoustic observation given each of the sub-word units. During training, model parameters are tuned such that this likelihood is maximized for the correct label sequence (given by the transcription). At test time, the model provides scores for each label given the feature sequence from the test utterance. A beam-search decode procedure is then followed to determine the best sequence of labels $\mathbf{l} = [l_1, l_2, \cdots, l_L]$ using the acoustic scores provided by the model. A lexicon can be used to constrain the beam search to only those label sequences which correspond to valid words. Furthermore, a language model can be used to rescore the different candidate label sequences according to their language probability. The contribution from the lexicon and the language model can be represented by a separate score $p(\mathbf{l})$ assigned to each label sequence which adjusts the original scores given by the acoustic model. The final output of the decoder can thus be expressed as,

$$\mathbf{l}^* = \arg\max_{\mathbf{l}} p(\mathbf{l}|\mathbf{x})p(\mathbf{l}). \tag{2.2}$$

It should be noted that the training data for ASR is almost always in the form of (waveform, transcription) pairs, and the time alignment information for the labels in a transcription is not available. Thus, the training procedure should have some form of internal alignment in addition to parameter tuning. In Section 2.3, we discuss how this alignment is done in different acoustic models.

2.2 The ASR front-end

The front-end is responsible for converting raw waveforms into feature vectors that are representative of the sound class, while rejecting other sources of variability as much as possible. Until recently, MFCCs were the standard feature type used in most ASR systems. However, the final Discrete Cosine Transform (DCT) in MFCC features was only intended

to accomodate the use of diagonal-covariance GMMs as acoustic models by decorrelating the different feature dimensions. Such decorrelation is no longer necessary with DNN acoustic models. In fact, DNNs are known to make use of such correlations in order to achieve better classification. As a result, Mel Filter-Bank (MFB) features are now the standard feature for ASR. The front-end can optionally include different enhancement strategies incorporated at different stages of the feature extraction pipeline. We review such enhancement strategies in Section 2.6.

2.3 Acoustic modeling

At the core of a speech recognition system is the statistical models assigned to different speech sounds which provide the likelihoods $p(\mathbf{l}|\mathbf{x})$ in Eq. (2.2) for recognition. Here we review three acoustic modeling techniques used in ASR.

2.3.1 GMM-HMM models

GMM-HMM models were a dominant acoustic modeling choice in ASR for many years before they were replaced with DNN-based alternatives. For smaller training datasets, they are still a useful choice. In GMM-HMM models for LVCSR, often a 3-state HMM is used to model the temporal evolution of different speech sounds, with a GMM characterizing the distribution of data at each hidden state. To be able to reliably train all GMM parameters, GMMs with diagnoal covariances are often used, which requires features with decorrelated dimensions (e.g., MFCCs). All GMM parameters are trained based on available training data using the well-known Baum-Welch algorithm [22].

To consider the context dependency of phonemes in speech, each HMM models a triphone (context-dependent phone with left and right contexts). The problem with using triphones is that there are a large number of possible triphones given a fixed set of monophones, and so the amount of training data for each triphone model will inevitably be small. To overcome

this problem, the parameters of different HMM states are often tied together using a decision tree clustering approach [23].

A limitation of HMM-based models is the simplifying dependency assumptions that are made about the data, which are not accurate about speech signals. The *conditional independence* assumption states that given the HMM internal state, the observation at each time step is independent of all past observations. Moreover, the first-order *Markovian chain* assumption implies that state occupancies are only determined based on the previous state. Another difficulty with HMM-based models is due to the lack of aligned labels for the acoustic features. The training data in ASR is pairs of speech signals with the corresponding label sequence, without any available alignment between them[1]. As a result, the training procedure should implicitly also align the labels with the features. There are two popular approaches to achieve this with GMM-HMM models. The first is to concatenate the HMMs of all labels in a certain training example, and train the resulting composite HMM as a whole using Baum-Welch algorithm [24]. Another approach is to initially assume a uniform alignment (where each label consumes a fixed number of features), and iteratively retrain and refine the alignment boundaries [25].

2.3.2 Hybrid DNN-HMM models

The first structure which made use of deep neural networks within the acoustic model was DNN-HMM hybrid models. These models use an HMM to model temporal dynamics (similar to conventional GMM-HMM models), but replace all the GMMs with a single feed-forward DNN which is responsible to predict posterior probability of each HMM state given the acoustic observation. In other words, the output space for the DNN is still defined by the decision-tree clustering performed for an initially trained separate GMM-HMM model.

[1]For phoneme labels, obtaining a manual alignment between the sequence of phonemes and a speech signal is too cumbersome and impractical for large datasets. In the case of text characters, there is often no unique alignment.

The DNN-HMM hybrids were shown to significantly outperform conventional GMM-HMMs on a variety of different benchmarks, datasets, and tasks [11]. As a result, they quickly replaced older GMM-based models and are now widely accepted as state-of-the-art in speech recognition.

2.3.3 End-to-end RNN models

In spite of the impressive results obtained from DNN-HMM hybrids, they are still dependent on a separate GMM-HMM model which defines the output space and provides senone labels for training. In other words, part of the learning is still dependent on hand-crafted and human-designed procedures such as phoneme dictionary specification, context-dependency definitions (biphones, triphones), decision-tree state clustering, etc. These rules define the set of classes for which a DNN classifier is built during training. Although the DNN can achieve better classification compared to GMMs, it is still forced to learn the fixed representations defined by the initial GMM-HMM model. Moreover, this implies that the HMM simplifying assumptions mentioned in Section 2.3.1 will also carry over to the DNN model.

An alternative to DNN-HMM hybrids which has quickly attracted a lot of attention is the so-called end-to-end models, in which the acoustic model consists of a single RNN, without any reliance on a separate HMM. The feedback paths in RNNs make them suitable for sequence modeling tasks without any additional mechanism to handle temporal dynamics. In other words, RNNs are capable of jointly modeling both temporal aspects and state distributions at the same time. Moreover, since RNNs directly map from a sequence of acoustic features to the corresponding sequence of labels, no explicit alignment is needed during training. End-to-end RNN models possess charactertistics which result in drastically simplified ASR pipelines:

- RNNs model context dependencies through their internal states, instead of requiring the output space to encode phoneme contexts. As a result, there is no need to define

context-dependent labels (e.g., triphones). RNN models have been shown to provide state-of-the-art performance with mono-phone output spaces, making the use of triphones and decision-tree state clustering unnecessary.

- Even further, RNN models can be used with character output spaces instead of phonemes. This is because RNN-based acoustic models are essentially sequence-to-sequence models which map a sequence of features to a sequence of labels, without any explicit alignment. Therefore, since no specific label is needed for individual frames, the output sequence can be any arbitrary chain of symbols which represents the content of the feature sequence. Using character output spaces has the significant advantage that it does not require a phoneme dictionary, which used to be the primary source of requirement for human expertise in conventional ASR systems.

- The number of senones in conventional HMM-based systems is on the order of thousands, resulting in a very large decode graph and slow beam search. The use of monophones or character output spaces in RNN acoustic models results in a much simpler decoding graph which enables faster decoding.

The above mentioned benefits come at the cost of requiring more training data. This is because end-to-end models are expected to learn everything from data, including temporal dynamics, alignment of features with labels, conversion to the character space, etc. In situations where such data is available, an end-to-end RNN model provides state-of-the-art results without requiring the conventionally designed complex ASR pipelines.

There are mainly two network structures which enable end-to-end training of RNN acoustic models, namely RNN-CTC models [26], and the more recent attention-based encoder-decoder models [27]. We briefly review both approaches here. However, end-to-end experiments in this dissertation are all based on the RNN-CTC approach.

RNN-CTC end-to-end models

The first (and currently most popular) neural network structure that enables end-to-end training is based on the Connectionist Temporal Classification (CTC) framework [26] for labeling unsegmented sequences. Figure 2.2 shows the basic architecture of RNN-CTC models. Given a sequence of feature vectors $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_T]$ from a speech utterance, a deep RNN applies multiple stages of nonlinear recurrent transformations of the form

$$\mathbf{h}_t^{(l)} = f(\mathbf{h}_{t-1}^{(l)}, \mathbf{h}_t^{(l-1)}, \boldsymbol{\theta}^{(l)}), \tag{2.3}$$

where $\mathbf{h}_t^{(l)}$ is the output of layer l at time frame t ($\mathbf{h}_t^{(0)} = \mathbf{x}_t$), $\boldsymbol{\theta}^{(l)}$ represents the trainable parameters of layer l, and $f(\cdot)$ is used to generally denote the internal layer transformations of the particular recurrent architecture that is used (either a Long Short-Term Memory (LSTM) layer [28] or a Gated Recurrrent Unit (GRU) layer [29]). The activations of the last hidden layer are passed to a final softmax layer of size $|S|$, where $S = \{s_1, \cdots, s_{|S|}, blk\}$ is the output symbol set (phonemes or characters plus an additional *blank* label) and $|\cdot|$ represents the cardinality of the set. The softmax outputs at each frame are interpreted as the posterior probability of observing each of the labels at that frame:

$$p(s_{i,t}|\mathbf{X}) = \frac{\exp\left(\mathbf{w}_i^T \mathbf{h}_t^{(L)}\right)}{\sum_{j=1}^{|S|} \exp\left(\mathbf{w}_j^T \mathbf{h}_t^{(L)}\right)}. \tag{2.4}$$

Here, $p(s_{i,t}|\mathbf{X})$ represents the probability of observing symbol s_i at time t given the input sequence, and $\mathbf{w}_i^T$ denotes the transpose of a column weight vector from the softmax layer. The extra symbol (blk) represents a blank or no output at a particular frame, which enables the network to appropriately align the input features with the label sequence. Note that since the recurrent layers are often bi-directional, the posteriors at each frame are conditioned on the whole input sequence.

The CTC objective is to maximize the overall probability of the ground-truth label sequence given the observed feature sequence using any possible alignment between them.

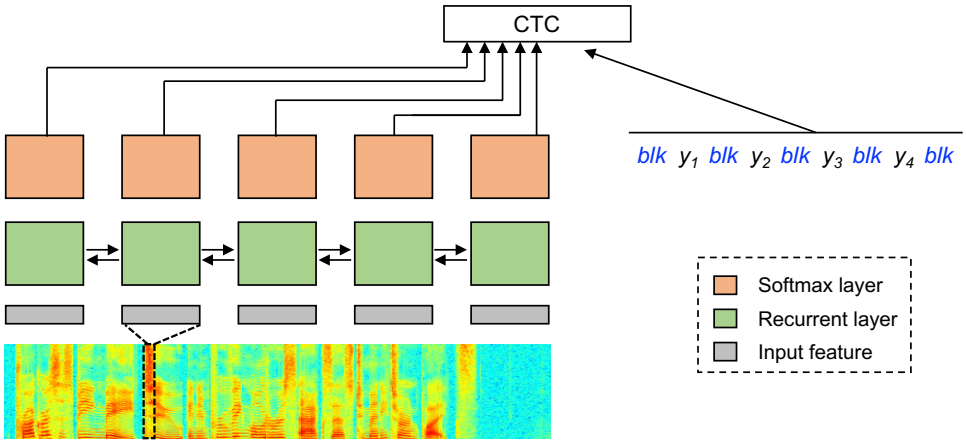


Figure 2.2. RNN-CTC acoustic models

This is obtained by summing over the probabilities given by all possible alignments:

$$J_{CTC} = -\log\Big(\sum_{a\in\mathcal{A}}\prod_{t=1}^{T} p\big(s_{a[t],t}|\mathbf{X}\big)\Big). \tag{2.5}$$

Here, $\mathcal{A}$ is the set of all possible alignments and $a[t]$ is one such alignment which gives a symbol index for every time frame t. The sum in (2.5) is efficiently computed using a dynamic programming forward-backward algorithm which makes use of the intermediate probabilities of partial label sequences [26]. The gradients resulting from the CTC objective are back-propagated through time and over all hidden layers to tune the network parameters. To decode a particular test utterance, the simplest approach is to use a memoryless search by selecting the most active output at each frame followed by removing blanks and label repetitions (referred to as best-path decoding). Alternatively, we can track multiple paths in a beam search algorithm similar to [30] to find the most likely label sequence. Moreover, to efficiently incorporate a language model and an orthographic lexicon, we can use a conventional Weighted Finite State Transducer (WFST) to construct a search graph similar to [31].

Implicit in Equation 2.5 is the assumption of conditional independence between the network *outputs* at different time steps given the internal hidden state of the network. This

assumption is valid since no feedback connections exist from the predicted outputs to the network layers. Note that this is a fundamentally different assumption from the conditional independece assumption in HMMs. While in HMMs the input features at different time steps are assumed independent, RNNs make no such simplifying assumption about the input sequence. Rather, the conditional independence in the CTC framework refers to the output symbols at different time steps. The consequence of this assumption is that the network will not be able to learn any information about common patterns in the output sequences, and will have to rely on a separate language model to incorporate sequential structure of the labels. In other words, the acoustic model and the language model in RNN-CTC systems are two completely disjoint modules whose scores should be properly combined prior to decoding.

Attention-based encoder-decoder models

An alternative structure for end-to-end acoustic modeling is sequence-to-sequence models, where an encoder RNN transforms the input sequence into a fixed-length compact representation, and a decoder RNN uses this representation to produce the sequence of output labels. These models were initially used in Neural Machine Translation (NMT) to map a sequence of words in one language to another [32], and were later found to be useful in acoustic modeling in ASR as well [27, 33].

Figure 2.3 shows the basic architecture of an attention-based RNN acoustic model. Given a sequence of acoustic features $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_T]$, the encoder RNN maps them to a synchronous sequence of higher level representations $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_T]$. At the k'th step of the decoder, first an attention model produces a *glimpse* vector $\mathbf{g}_k$ using a weighted average of the encoded features

$$\mathbf{g}_k = \sum_{t=1}^{T} w_{k,t} \mathbf{h}_t. \tag{2.6}$$

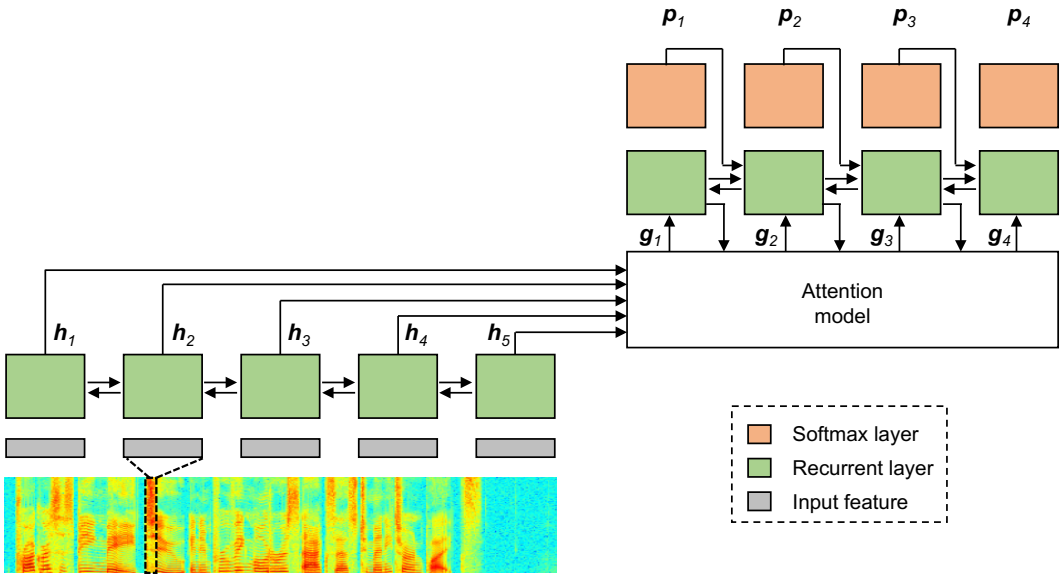


Figure 2.3. attention-based encoder-decoder acoustic models

The weights $w_{k,t}$ in (2.6) determine the contribution of t'th frame of the speech signal on the k'th produced label by the decoder, and are therefore refered to as an *alignment*. The role of the attention model is to provide this alignment given the previous hidden state of the decoder and the elements of $\mathbf{h}$:

$$w_{k,t} = f_{Attend}\big(\mathbf{q}_{(k-1)}, \mathbf{h}_t\big). \tag{2.7}$$

The glimpse vector, the previous hidden decoder hidden state, and the previous output are all used to complete the decoder's recurrence step, and a final softmax layer predicts the label posterior distribution at each step:

$$\mathbf{q}_k = f_{RNN}(\mathbf{g}_k, \mathbf{q}_{k-1}, \mathbf{p}_{k-1}), \tag{2.8}$$

$$\mathbf{p}_k = [p_0(k), \cdots, p_{|S|}(k)] = f_{out}(\mathbf{q}_k). \tag{2.9}$$

Here, $p_i(k)$ is the probability of outputting symbol s_i at the k'th step of the decoder. The parameters of the encoder, decoder, and the attention model are jointly optimized based on the average cross-entropy error between the decoder outputs and the ground-truth label

sequence $[y_1, \cdots, y_L]$:

$$\boldsymbol{\theta} = \arg\max \frac{1}{L} \sum_{k=1}^{L} \log p_{y_k}(k). \tag{2.10}$$

Here, $\boldsymbol{\theta}$ represents the set of all parameters in the network, and $p_{y_k}(k)$ is the predicted posterior probability for the ground-truth label y_k at time step k.

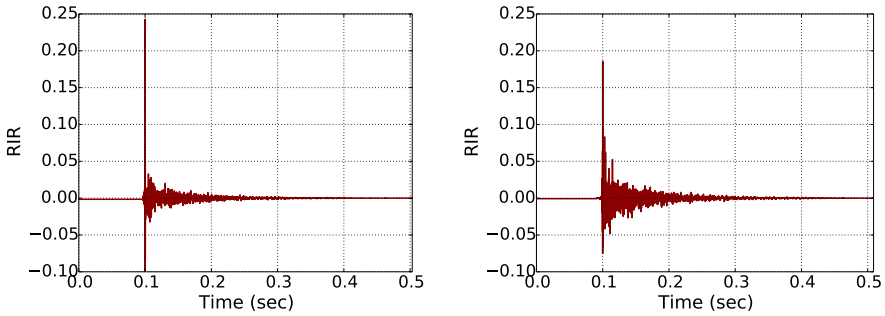
A major difference between this model and the RNN-CTC model is the inclusion of the previous output $\mathbf{p}_{k-1}$ in the decoder's recurrence, which allows it to learn information about output sequences. In RNN-CTC models, outputs at different time frames are assumed conditionally independent. The model is thus confined to learn only the acoustic information about the input sequence, requiring a separate language model to adjust acoustic scores prior to decoding. Here, the model is allowed to learn both acoustic and sequential (language) information. As a result, when a separate language model is not provided, the attention approach often outperforms RNN-CTC models.

2.4 Distant speech model

The effect of sound reflections received by a distant microphone in a reverberant environment can be modeled as a convolution between the clean speech signal and a Room Impulse Response (RIR) which characterizes the acoustic path from the speaker to the microphone:

$$x(n) = \sum_{q=0}^{L_h} h(q)s(n-q) + e(n). \tag{2.11}$$

Here, $s(n)$ is the clean speech, $x(n)$ is the received microphone signal, $e(n)$ is the additive environment or recording noise, and $h(n)$ is the RIR. Figure 2.4 shows two example RIRs from an office room with a reverberation time of $T_{60} \approx 430\ ms$. RIRs are typically divided into three regions, namely the *direct path* which is the desired component, *early reverberation* which is a number of distinct reflections arriving soon after the direct path (typically within 50 msec), and *late reverberation* which is a build-up of thousands of reflections which are not



(a) d=1 m (b) d=3 m

Figure 2.4. Example RIRs from an office room with $T_{60} \approx 430\ ms$ (from Aachen Impulse Response (AIR) dataset [1]).

clearly distinguishable from each other. The reverberation model in (2.11) can be rewritten based on these three components:

$$x(n) = s_d(n) + e'(n), \tag{2.12}$$

$$s_d(n) = h_d(n) * s(n), \tag{2.13}$$

$$e'(n) = h_e(n) * s(n) + h_l(n) * s(n) + e(n), \tag{2.14}$$

where $h_d(n)$, $h_e(n)$ and $h_l(n)$ represent the direct path, early reverberation and late reverberation, respectively, and $*$ denotes convolution. The relative energy of the direct path compared with the reflections is known as direct-to-reverberation ratio (DRR) and has a high correlation with ASR accuracy. The value of DRR is a function of reverberation time and the speaker-to-microphone distance. A longer reverberation time leads to more reverberant energy and lower DRR. Also, a longer speaker-to-microphone distance results in lower direct-path energy while the late reverberation energy stays the same, hence resulting in a lower DRR. Moreover, the effects of speaker head orientation can also be subsumed in the RIR. A speaker who is not facing a microphone emits less energy in the direct path component, effectively reducing DRR.

It should be noted that even in the absence of environmental noise (i.e., when $e(n) = 0$), the interference term $e'(n)$ is a non-stationary, non-Gaussian and colored noise which

is correlated with the desired clean component $s_d(n)$. This is a very challenging type of interference to remove, as most noise-robust approaches assume uncorrelated and stationary noise. As a result, conventional noise-robust approaches that are designed to address additive noise are often not useful to deal with the reverberation problem in distant ASR.

The RIR length is a function of room reverberation time (T_{60}) and varies from a few hundred miliseconds (e.g., typical office rooms) to a few seconds (e.g., highly reverberant auditoriums, etc.). In almost all practical cases, the RIR length is larger than the typical frame length used in speech recognition (25-30 msec). As a result, the convolutive relation in (2.11) cannot be described as a simple multiplicative relation in the frequency domain. This is what makes far-field robustness a more challenging problem compared to other variability compensations in ASR (noise and channel robustness, speaker independence, etc.), where distortions are limited to multiplicative or additive effects in the feature domain. The time domain model in (2.11) is described in the Short Time Fourier Transform (STFT) domain as

$$\tilde{X}(k,m) = \sum_{k'=0}^{K-1} \sum_{p=0}^{L-1} \tilde{H}_{kk'}(p)\tilde{S}(k', m-p) + \tilde{E}(k,m), \tag{2.15}$$

where k is the frequency bin index, m is frame index, $\tilde{S}(k,m)$ and $\tilde{X}(k,m)$ are the complex STFT coefficients of the clean speech and received microphone signal, $\tilde{E}(k,m)$ is the STFT of the noise, and $\tilde{H}_{kk'}(m)$ is a time-frequency representation of the RIR using an analysis window of the form,

$$w_{kk'}(n) = w_a(n)e^{j\frac{2\pi}{N}kn} * w_s(n)e^{j\frac{2\pi}{N}k'n}, \tag{2.16}$$

where $w_a(n)$ and $w_s(n)$ are the analysis and synthesis windows used in the STFT of the input speech, N is the Discrete Fourier Transform (DFT) length, and $*$ denotes convolution. Using the window given in (2.16), the time-frequency representation $H_{kk'}(m)$ can be expressed as

$$H_{kk'}(m) = \sum_n h(n)w_{kk'}(mB - n), \tag{2.17}$$

where B is the skip period in STFT analysis, and the summation is over the length of the window. The cross-band filters $H_{kk'}(m)$ $(k \neq k')$ represent the aliasing effects due to the downsampling operation inherent in a STFT analysis with skip period of $B > 1$. In other words, the contents of each frequency band in the STFT of reverberant speech is influenced by adjacent bands in the clean speech. However, the energy of the cross-band filters $H_{kk'}(m)$ (for $k \neq k'$) are small compared to the band-to-band filters $H_{kk}(m)$ due to the increasingly smaller overlap regions between the two modulated windows in (2.16) as $|k - k'|$ increases. As a result, it is common practice to consider the cross-band terms as additional noise terms, and write (2.15) in the magnitude STFT domain as

$$X(k,m) = \sum_{p=0}^{L-1} H(k,p)S(k,m-p) + E(k,m), \tag{2.18}$$

where $S(k,m)$ and $X(k,m)$ are the magnitude STFTs of the clean speech and recorded microphone signal, $H(k,m) = |\tilde{H}_{kk}(m)|$ represents the spectral envelope of the RIR from the speaker position to the microphone, and $E(k,m)$ represents the combined effects of additive noise, cross-band terms, and the error introduced by replacing the magnitude of the sum as required by (2.15) by a sum of magnitudes. Similar to $h(n)$, the length of the filters $H(k,m)$ is a function of room reverberation time. The relationship in (2.18) is a very useful representation which explicitly models the masking effect of reverberation in each frequency band (using the filters $H(k,m)$), and serves as a basis for many studies on reverberant speech [34–36]. The zero-lag coefficients ($H(k,0)$) model the *self-masking* effects of reverberation, which are the temporal smearing distortions that occur within a single frame. The other filter coefficients ($H(k,p), p = 1, \cdots, L-1$) represent *overlap-masking*, which is the long-term smearing effects from each frame on the subsequent frames.

2.5 Fundamental problem in distant speech recognition

Fig. 2.5 illustrates a comparison between a clean speech signal and its reverberant version recorded at a distance of 3 meters. It can be seen that reverberation causes a *temporal*

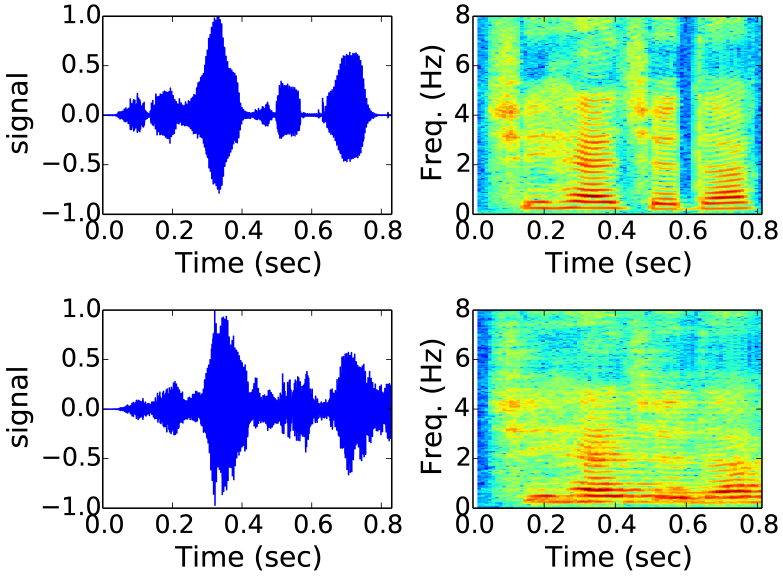


Figure 2.5. Waveform and spectrogram comparison between clean and reverberant speech for an office room with $T_{60} \approx 430\ ms$ and at a distance of $d = 3\ m$.

smearing effect, increasing long-term correlations in the sequence of spectral features derived from the short-term frames. This can be viewed as a leakage of spectral content from each frame to subsequent frames in each frequency band, which is also indicated by the model in (2.18). In other words, unlike other robustness issues such as additive noise, speaking style or microphone mismatch, reverberation does not independently affect each frame. Rather, it has a long-term effect which spans multiple time frames.

The long-term dependencies in far-field speech are specifically detrimental to the operation of Hidden Markov Models (HMMs). Both of the underlying assumptions in HMM acoustic models, namely the conditional independence and the first order Markov assumption, are inaccurate for reverberant speech. Using these assumptions, HMMs describe the likelihood of the observed sequence of speech features as

$$p(\mathbf{x}_1 \cdots \mathbf{x}_T) = \sum_{\mathbf{s}} \prod_{t=1}^{T} p(\mathbf{x}_t|s_t)p(s_t|s_{t-1}), \tag{2.19}$$

where $\mathbf{x}_t$ and s_t represent the speech feature vector and the internal state at time t, and the summation is over all possible state sequences $\mathbf{s} = [s_1, \cdots, s_T]$. Based on Equation (2.19),

once the current state is determined, the distribution is fixed and independent of past visited states. This does not match the described far-field speech model in (2.18), where each frame is influenced by multiple frames in the past, and the distribution is directly influenced by these past observations. Therefore, a solution for far-field ASR must inevitably consider past observed features and states in order to specify the probability distribution for the current frame (i.e., it should describe probabilities of the form $p(\mathbf{X}_t|\mathbf{X}_1, \cdots, \mathbf{X}_{t-1}, s_1, \cdots, s_t)$).

Many front-end enhancement strategies attempt to alleviate this problem by incorporating longer context into the feature processing stage. This includes using delta or double-delta features, feature concatenation across multiple time frames, log-spectral mean and variance normalization, etc. However, these are not sufficient to provide robust predictions due to the highly non-stationary nature of the distortions in a far-field signal. Log-spectral mean normalization, in particular, is intended to remove transient (fast-decaying) channel distortions that are additive in the log-spectral domain, and can thus remove only the effects of early reverberation (self-masking). Any similar feature normalization strategy which applies the same normalizing function to all frames in an utterance is unable to address the late reverberation problem due to the context-dependent nature of the distortion.

2.6 Existing solutions for distant speech recognition

2.6.1 Microphone array processing

Using an array of closely-spaced microphones to capture a higher-quality signal has by far been the most popular solution to the problem of distant speech recognition [37–40]. Such uniform microphone arrays enable beamforming, which results in a signal with considerably improved SNR and DRR. A comprehensive review of microphone array techniques can be found in [41].

2.6.2 Robust features

There have been many studies which explore feature representations that are inherently robust to reverberation. These include RASTA-PLP [42] or MHEC [43] features. Based on the discussion in Section 2.5, these methods are insufficient to address the reverberation problem due to the cross-frame correlations involved. These approaches can at best alleviate only the self-masking effects of early reverberation, and they are unable to compensate the overlap-masking effects of late reverberation.

2.6.3 Feature enhancement

Many existing solutions use single-channel enhancement, aiming to recover the clean features by jointly processing the sequence of corrupted observations from a single microphone. These approaches can be broadly categorized to three groups according to where the enhancement takes place in the front-end processing pipeline. *Blind deconvolution* approaches enhance the time domain signal or the complex STFT coefficients to estimate a clean waveform, which is input to a standard feature extraction pipeline [44–46]. *Spectral enhancement* approaches, on the other hand, aim at suppressing noise and reverberation in the power spectrum domain. This enhancement can either be carried out on the full-resolution magnitude STFT coefficients [34, 35], or directly on the final log-mel spectral or cepstral features [47, 48].

2.6.4 Model adaptation

Model adaptation aims at tuning the parameters of an existing acoustic model to better match a set of far-field observations. There are two different scenarios in which adaptation is useful. The first is the adaptation of a clean-trained acoustic model to a set of far-field observations. Here, the original acoustic model is trained on a large dataset of clean (close-talking) features. The distribution of different phoneme classes can be learned more easily and effectively in this setting, because there is no context-dependent distortion in the

training data. However, the clean-trained model cannot be directly used on mismatched far-field recordings. Starting from this clean-trained model and using a dataset of transcribed far-field observations, we can adapt the model parameters to reduce the mismatch with far-field data. The second scenario for adaptation is when the original model is already trained on multi-condition far-field data from a variety of different room characteristics. The original model in this scenario has less mismatch with the test data, but its paramters have been optimized to provide *average best performance* across the room conditions present in the training set. If a few transcribed sentences are available from a specific target environment, model adaptation can help improve the performance for that particular environment.

2.6.5 Multi-condition training

If a large corpus of far-field speech recorded in reverberant environments with different acoustic properties (reverberation time, DRR, etc.) is available, we can train acoustic models directly based on far-field features without any front-end compensation. Although this is a more difficult learning problem because of the distortions in training data, it often results in large improvements compared to most front-end compensation approaches due to the significantly reduced mismatch between training and test conditions. RNN models are in particular an attractive choice for multi-condition training with reverberant data due to their remarkable capability to learn long-term correlations caused by reverberation. A deep RNN that is trained on a sufficiently large dataset of reverberant speech often provides very competetive accuracies which outperforms most manually designed enhancement approaches.

CHAPTER 3

A ROBUST MULTICHANNEL FRONT-END FOR DISTRIBUTED MICROPHONE ARRAYS: CNTF ALGORITHM

xThis chapter introduces CNTF algorithm[1], a front-end enhancement solution that is designed to improve reverberation robustness in distributed microphone arrays. Microphone array processing has been a major solution for obtaining better quality recordings in noisy and reverberant environments [37–40]. A conventional microphone array consists of a few closely-spaced microphones with a fixed known geometry and shared processing among the elements (common clock pulse). By sampling the sound field from multiple known locations, a microphone array enables us to perform spatial filtering (in the form of fixed or adaptive beamforming) to enhance the desired signal coming from a target location. Although such beamforming techniques are one of the most common solutions for distant speech recognition, there are a number of factors which limit their applicability. We first discuss these limitations and then focus on the more flexible case of distributed arrays, where the different channels are independent recording devices in random unknown locations.

3.1 Limitations of conventional microphone arrays

In spite of their popularity as a front-end enhancement solution for distant ASR, conventional array processing approaches that depend on inter-element phase information have the following limitations:

- Microphone array approaches necessarily need uniform configurations of closely-spaced microphones that are designed and calibrated in advance. In particular, all array

[1] © 2016 IEEE. Reprinted with permission, from S. Mirsamadi and J. H. L. Hansen, A Generalized Nonnegative Matrix Factorization Approach for Distant Speech Recognition with Distributed Microphones, IEEE Trans. Audio Speech Lang. Process., vol. 24, no. 10, Oct. 2016.

elements must be driven by the same processor and clock pulse, so that they share the same time reference and sampling frequency.

- Microphone arrays require knowledge of speaker location. Although this information can be estimated from the array signals, sound source localization in reverberant environments is itself a challenging task which inevitably introduces errors.

- A microphone array's look direction is not narrow enough to cancel all reflections. Although there are techniques such as filter-and-sum to arbitrarily design an array's directional response [41], there is a limit on how narrow the mainlobe width can be designed (enforced by the array geometry). This is usually not a problem for cancelling additive interferences unless the noise source happens to be in the same direction as the speaker. However, with reverberation, there are always many reflections arriving from all directions including the array's look direction. As a result, beamforming is fundamentally a sub-optimal approach for achieving reverberation robustness.

3.2 Distributed microphone arrays

In this study, as an alternative to classical array processing, we consider situations where we have multiple microphones available, but they do not form a compact synchronous array. Instead, they are independent recording devices distributed in random unknown locations, and there is no assumed synchrony between their signals (they do not share the same clock). This is a more flexible situation that covers a wider range of applications (it eliminates the need to have pre-designed and calibrated arrays), but it gives rise to a number of new challenges which makes conventional techniques inapplicable [49, 50]. First, the lack of synchrony among the different channels means there is no meaningful time-delay information between them, hence making beamforming impractical. The delays between signals of different recording devices in this case are directly dependent on segmentation decisions which

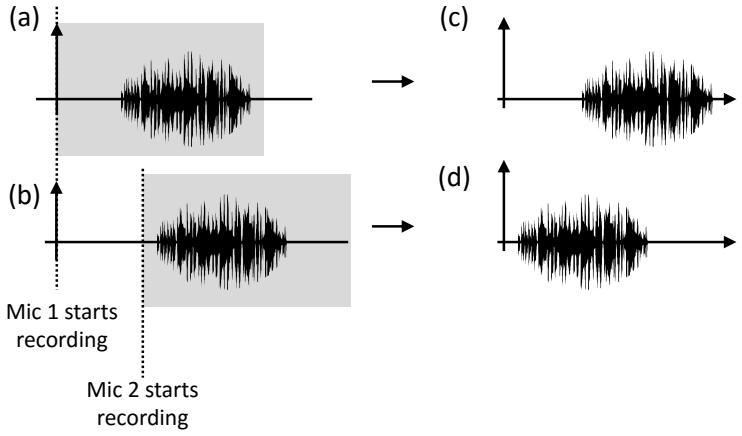


Figure 3.1. An illustration of the problem of no shared time reference in distributed audio processing (gray areas indicate the recorded segments of audio). (a) Sound wave reaching microphone 1; (b) Sound wave reaching microphone 2 (delayed with respect to microphone 1); (c) Recorded waveform by microphone 1; (d) Recorded waveform by microphone 2 (falsely indicating a time-advance with respect to microphone 1);

mark the beginning of an utterance. Thus, they do not carry any meaningful spatial information. Fig. 3.1 illustrates this problem for two distributed microphones. While the sound wave reaches microphone 2 with a delay compared to microphone 1, the recorded waveforms falsely show the opposite, because there is no shared time reference. Another challenge with distributed arrays is the possibility of very different SNRs or DRRs among the various channels. The quality of a recording channel is directly dependent on how close it is to the source. Thus, unlike compact uniform arrays, it is very common to have significantly different signal qualities among the different channels.

The solution in such distributed array scenarios is to combine information from the different channels in the power spectrum (or magnitude STFT) domain. According to the far-field model in 2.18, the subband magnitudes in this case are different convolutively distorted versions of the the same subband in the clean speech. By jointly processing such distorted power spectra from multiple channels, we can obtain an estimate of the clean speech component which is shared among them, and discard the differences as the contribution from the corresponding RIRs. Another advantage is that power spectrum enhancement is less

sensitive to speaker movements, because the spectral envelope of the reverberation tail is fairly insensitive to the precise locations of the source and microphone.

3.3 CNTF algorithm

This section describes our proposed convolutive nonnegative tensor factorization (CNTF) algorithm for reverberation-robust feature extraction in distributed microphone arrays. We first describe a non-negative tensor model for multi-channel spectra in reverberant environments, and then describe how this tensor can be decomposed into a clean speech estimate and the corresponding RIR elements.

3.3.1 Nonnegative tensor model for reverant spectrograms

Recall from Section 2.4 that the following relationship holds between the magnitude STFT coefficients of each channel and the original clean speech:

$$X^{(i)}(k,m) = \sum_{p=0}^{L-1} H^{(i)}(k,p) S(k,m-p) \tag{3.1}$$

where $S(k,m)$ and $X^{(i)}(k,m)$ are the magnitude STFTs of the clean speech and reverberant speech of the i'th channel, and $H^{(i)}(k,m)$ represents the spectral envelope of the RIR from the speaker position to the i'th microphone. The superscript (i) is used to represent the channel index throughout this chapter.

The magnitude spectrogram matrices from the individual channels can be viewed as frontal slices of a third-order nonnegative tensor $\mathcal{X}$ with dimensions $K \times M \times C$, where K is the number of frequency bins in the STFT analysis, M is the total number of frames in the utterance, and C is the number of microphones. Based on the signal model in (3.1), the frontal slices of this tensor can be obtained by multiplying delayed versions of the source signal's spectrogram matrix by different diagonal matrices representing the different taps of

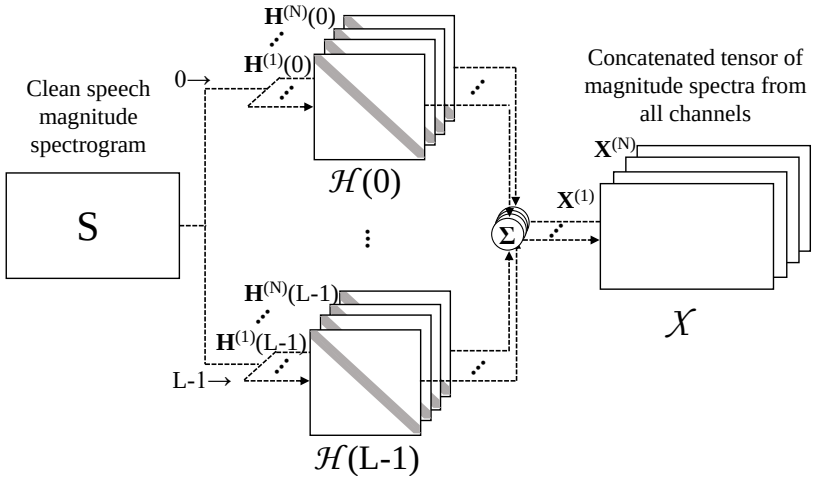


Figure 3.2. Convolutive tensor model for multichannel reverberant speech recognition. $\mathcal{X}$ and $\mathcal{H}(p)$ are used to denote three-dimensional tensors as a whole, and the matrices $\mathbf{X}^{(i)}$ and $\mathbf{H}^{(i)}(p)$ represent the frontal slices of these tensors, i.e. the magnitude spectrogram and the RIR component for the i'th channel.

the RIR spectral envelopes:

$$\mathbf{X}^{(i)} \approx \sum_{p=0}^{L-1} \mathbf{H}^{(i)}(p)\mathbf{S}^{p\rightarrow}. \tag{3.2}$$

Here, $\mathbf{H}^{(i)}(p)$ is a $K \times K$ nonnegative diagonal matrix whose diagonal elements are $H^{(i)}(p,k)$ $(k = 0, \cdots, K-1)$, $\mathbf{S}$ is the spectrogram matrix of the clean speech, and the operator $p\rightarrow$ shifts the rows of its argument matrix by p positions to the right, filling in zeros from the left. Multiplying these shifted spectrograms with diagonal matrices $\mathbf{H}^{(i)}(p)$ and adding the results is equivalent to the individual convolutions in each frequency band. Note that $\mathbf{H}^{(i)}(p)$ is assumed to be diagonal due to the approximate model in (3.1) which does not use the cross-band filters in 2.15. Fig. 3.2 illustrates the described tensor model.

The introduced tensor model enables us to remove reverberation by using a convolutive extension of nonnegative tensor factorization (NTF) [51] in order to decompose $\mathcal{X}$ into a sum of tensor-matrix products, with the frontal slices of the factor tensors constrained to be diagonal. The resulting algorithm will thus attempt to discover a common component between the frontal slices of $\mathcal{X}$, which is the clean speech spectrogram, and discard the differences as the contribution from the convolutive effects of $\mathbf{H}^{(i)}(p)$ (i.e., the RIR effects).

To achieve this decomposition, a divergence measure is minimized between the observed tensor $\mathcal{X}$ and its estimate $\mathcal{Z}$ given by the nonnegative factors. The objective function for the CNTF algorithm can thus be expressed in terms of the tensor elements as,

$$J = \sum_{i,k,m} D\big[X^{(i)}(k,m)||Z^{(i)}(k,m)\big], \tag{3.3}$$

where

$$Z^{(i)}(k,m) = \sum_{p=0}^{L-1} \hat{H}^{(i)}(k,p)\hat{S}(k,m-p). \tag{3.4}$$

Here, $\hat{H}^{(i)}(k,p)$ and $\hat{S}(k,m-p)$ represent the current estimates of the nonnegative factors. Note that since the base matrices $\mathbf{H}^{(i)}(p)$ in (3.2) are diagonal, the optimization can be carried out independently in each frequency bin k by operating on the scalars $\hat{H}^{(i)}(k,p)$ and $\hat{S}(k,p)$ instead of the entire matrices in (3.2). The remainder of the derivations here will therefore follow this scalar form.

3.3.2 Alpha-beta divergence

The choice of the divergence measure for tensor factorization influences the performance of the algorithm because it is closely related to the distribution of the data. Most common choices in standard nonngetive matrix factorization (NMF) are the Euclidean Distance (ED) and the (generalized) Kullback-Leibler (KL) divergence [52], although other divergence measures have also been studied [53, 54]. Each of these divergences correspond to a certain underlying generative model assumed for the input data through the following relationship (here we drop channel, frame, and frequency indices for simplicity):

$$p(X|Z;\boldsymbol{\theta}) = \frac{1}{f(\boldsymbol{\theta})} \exp\big(-D(X\|Z;\boldsymbol{\theta})\big), \tag{3.5}$$

where $D(X||Z;\boldsymbol{\theta})$ indicates a divergence measure with parameters $\boldsymbol{\theta}$, and $f(\boldsymbol{\theta})$ is a normalizing factor (called the partition function) which makes $p(X|Z;\boldsymbol{\theta})$ a valid probability density function:

$$f(\boldsymbol{\theta}) = \int_X \exp(-D(X\|Z;\boldsymbol{\theta}))dX. \tag{3.6}$$

Table 3.1. common divergence measures as special cases of alpha-beta divergence.

α	β	divergence	underlying distribution
1	1	Euclidean Distance	Gaussian
1	0	Generalized Kullback-Leibler (KL) Divergence	Poisson
1	-1	Itakura-Saito (IS) divergence	Gamma

Based on this interpretation, minimizing a divergence measure is actually a maximum-likelihood (ML) estimation of the nonnegative factors assuming the corresponding distribution given by (3.5) [55]. For example, NMF with the Euclidean distance measure gives ML estimates for the factors under a Gaussian assumption for the input matrix elements (several other correspondences of this kind are listed in Table 3.1).

In this study, we propose to use the general family of $\alpha\beta$-divergences [56] for the described tensor factorization task in multichannel dereverberation:

$$D_{\alpha\beta}(X\|Z) = -\frac{1}{\alpha\beta}(X^{\alpha}Z^{\beta} - \frac{\alpha}{\alpha+\beta}X^{\alpha+\beta} - \frac{\beta}{\alpha+\beta}Z^{\alpha+\beta}). \tag{3.7}$$

It has been shown in [56] that $D_{\alpha\beta}(X\|Z)$ in Equation (3.7) is a valid divergence measure which is nonnegative for all values of the arguments and has a global minimum of zero only when $X = Z$. This general definition can be extended by continuity to include the singularity points $\alpha = 0$, $\beta = 0$ and $\alpha + \beta = 0$, so that the divergence will be defined for all $\alpha, \beta \in \mathbb{R}$. These continuity extensions (as well as some other particular values for α and β, as summarized in Table 3.1), coincide with a number of popular divergence measures used in the literature. The alpha-beta divergence is thus a unifying generalization which interpolates between these measures, providing increased flexibility to match the actual data distribution according to (3.5) by appropriately selecting the parameters α and β. We will show in Section 3.5 that the use of $\alpha\beta$-divergence with appropriate values of the parameters improves performance of the CNTF algorithm, resulting in better ASR accuracy. This is

because the Gaussianity assumption underlying the commonly used Euclidean distance is inaccurate for the observed power spectrum data, which results in sub-optimal estimates for the factors.

Using $\alpha\beta$-divergence, our goal in CNTF enhancement is to find the factors $H^{(i)}(k,p)$ and $S(k,m)$ which minimize the cost function in (3.3) subject to the nonnegativity constraints:

$$H^{(i)}(k,p) \geqslant 0 \text{ and } S(k,m) \geqslant 0, \qquad \text{for all } i,k,p,m. \tag{3.8}$$

At the point of minimum divergence, $S(k,m)$ is expected to be an estimate the clean speech magnitude spectrum, while $H^{(i)}(k,m)$ will represent the RIR spectral envelope for channel i.

3.3.3 Multiplicative update rules

In this section, we derive update rules for $S(k,m)$ and $H^{(i)}(k,p)$ which yield the minimization of (3.3). As explained in Section 3.3.1, the diagonality of the frontal slices of tensors $\mathcal{H}(p)$ allows us to carry out the optimization independently in each frequency bin, thus breaking the matrix formulations into equivalent scalar forms with simpler gradient expressions.

For alpha and beta values in the region $\frac{1-\beta}{\alpha} \in [0,1]$, the cost function (3.3) is ensured to be convex with respect to either $\hat{S}(k,m)$ or $\hat{H}(k,p)$, but not both [56]. Consequently, the optimization of the nonnegative components are carried out in an alternating fashion, i.e., by keeping $\hat{S}(k,m)$ constant and updating $\hat{H}(k,p)$, and vice versa.

The derivatives of the cost function (3.3) with respect to the factor variables are:

$$\begin{aligned}\frac{\partial J}{\partial H^{(i)}(k,p)} = \frac{-1}{\alpha} \sum_{m} \big[& X^{(i)}(k,m)^{\alpha} Z^{(i)}(k,m)^{\beta-1} \\ & - Z^{(i)}(k,m)^{\alpha+\beta-1} \big] S(k,m-p),\end{aligned} \tag{3.9}$$

$$\begin{aligned}\frac{\partial J}{\partial S(k,l)} = \frac{-1}{\alpha} \sum_{i} \sum_{m} \big[& X^{(i)}(k,m)^{\alpha} Z^{(i)}(k,m)^{\beta-1} \\ & - Z^{(i)}(k,m)^{\alpha+\beta-1} \big] H^{(i)}(k,m-l).\end{aligned} \tag{3.10}$$

Gradient descent updates using the above derivatives does not necessarily preserve nonnegativity of the results. However, similar to conventional NMF algorithms [52], we can derive multiplicative update rules (which guarantee the preservation of nonnegativity) by assuming the following adaptive learning rates for the gradient descent optimizations:

$$\eta_H = \frac{H^{(i)}(k,p)}{\frac{1}{\alpha}\sum_m Z^{(i)}(k,m)^{\alpha+\beta-1} S(k,m-p)}, \tag{3.11}$$

$$\eta_S = \frac{S(k,l)}{\frac{1}{\alpha}\sum_i \sum_m Z^{(i)}(k,m)^{\alpha+\beta-1} H^{(i)}(k,m-l)}. \tag{3.12}$$

Using (3.11) and (3.12) with the gradients in (3.9) and (3.10) results in the following gradient descent update equations:

$$H^{(i)}(k,p) \leftarrow H^{(i)}(k,p)\frac{\sum_m Y^{(i)}(k,m) S(k,m-p)}{\sum_m Z^{(i)}(k,m)^{\alpha+\beta-1} S(k,m-p)}, \tag{3.13}$$

$$S(k,l) \leftarrow S(k,l)\frac{\sum_i \sum_m Y^{(i)}(k,m) H^{(i)}(k,m-l)}{\sum_i \sum_m Z^{(i)}(k,m)^{\alpha+\beta-1} H^{(i)}(k,m-l)}, \tag{3.14}$$

where,

$$Y^{(i)}(k,m) = X^{(i)}(k,m)^{\alpha} Z^{(i)}(k,m)^{\beta-1}. \tag{3.15}$$

Note that both numerators and denominators in update equations (3.13) and (3.14) are in the form of correlations between the estimated factors $\big(H^{(i)}(k,p)$ and $S(k,m)\big)$ and the intermediate variables $\big(Z^{(i)}(k,m)$ and $Y^{(i)}(k,m)\big)$. In practice, similar to [57], these correlations are computed via FFT multiplication which considerably reduces the computational complexity of the CNTF algorithm. Also note that by setting $\alpha = \beta = 1$ in (3.13) and (3.14), we obtain the update equations of [58] for CNTF with Euclidean Distance objective function. Moreover, by setting the number of channels to C=1, the algorithm simplifies to the single-channel CNMF algorithms of [34] and [59].

To address the scale indeterminacy inherent in the decomposition of (3.4), we impose the additional constraint $\sum_i \sum_p H^{(i)}(k,p) = 1$, which is satisfied by performing the following normalization for the RIR spectral envelopes after each update:

$$H^{(i)}(k,p) \leftarrow \frac{H^{(i)}(k,p)}{\sum_i \sum_p H^{(i)}(k,p)}. \tag{3.16}$$

We prefer to use this normalization strategy which is shared among the RIR components of different channels instead of individually normalizing $H^{(i)}(k,p)$ for each channel (i.e. normalizing only over different lags $p = 1, \cdots, L-1$ for each channel i). This is because the normalization in (3.16) allows the algorithm to automatically adjust the gains of the filters $H^{(i)}(k,p)$ according to the DRR of the corresponding channel. This will in turn adjust the contribution of each individual channel to the final estimate of $S(k,m)$ according to the distance between the corresponding microphone and the speaker. This is required in a blind scenario where we have no information about speaker or microphone locations. We believe such a strategy is beneficial compared to channel selection algorithms which identify noisy channels and completely eliminate their contribution [60, 61] (see the results in Section3.5.3 for more details about this automatic adjustment of channel contributions).

3.4 Parameter selection by score-matching

The update formulas provided in Section 3.3.3 can be used with the specific values of α and β listed in Table 3.1 which correspond to known divergences and distributions, as well as any other values which satisfy the convexity condition $\frac{1-\beta}{\alpha} \in [0,1]$. If a development dataset is available from the target environment, the best values for α and β can be selected based on closed-loop ASR performance on the development data. In this section, we describe an alternative method for automatically selecting the optimum values of alpha and beta for cases where a transcribed development dataset is not available.

As noted in Section 3.3.2, divergence measures correspond to likelihood assumptions for the data. Thus, by using $\alpha\beta$-divergence, we are assuming the distribution given in (3.5) for the magnitude STFT values, parametrized by $\boldsymbol{\theta} = [\alpha, \beta]$. With this assumption, maximum likelihood seems to be a natural choice to estimate the parameters α and β. However, this is not possible in general for $\alpha\beta$-divergence, since the partition function in (3.5) is analytically intractable and very difficult to compute numerically, except for the specific values of α and

β in Table 3.1. In [62], an alternative method called score-matching, is introduced for the estimation of such non-normalized models in which the distribution is only known up to a multiplicative constant ($f(\alpha, \beta)$ in our case). The score matching technique was extended to the case of nonnegative data in [63], and was successfully used for divergence selection in the task of music analysis in [64]. Here, we apply the score-matching technique for selecting the parameters of $\alpha\beta$-divergence in the CNTF algorithm.

3.4.1 The score-matching principle

The score function of a distribution is defined as the derivative of its log-density:

$$\psi(X;\boldsymbol{\theta}) = \frac{\partial \log p(X;\boldsymbol{\theta})}{\partial X}. \tag{3.17}$$

The point in using the score function is that it does not depend on the normalization term $f(\boldsymbol{\theta})$ (the partition function). It has been shown in [62] that the parameters of non-normalized models can be estimated by minimizing the expected distance between the score function resulting from the observed data and the score function given by the model. An extension of this principle for nonnegative data [63] states that the score-matching (SM) estimator for the parameters $\boldsymbol{\theta}$ is given by:

$$\boldsymbol{\theta}_{SM} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{SM}(\boldsymbol{\theta}), \tag{3.18}$$

$$\mathcal{L}_{SM}(\boldsymbol{\theta}) = \int_{\mathbb{R}^+} p(X;\boldsymbol{\theta}) \big[2X\psi(X;\boldsymbol{\theta}) + X^2\xi(X,\boldsymbol{\theta}) + \frac{1}{2}\psi^2(X;\boldsymbol{\theta})X^2\big] dX, \tag{3.19}$$

where $\xi(X;\boldsymbol{\theta})$ indicates the derivative of the score function (i.e., the second derivative of the log-density):

$$\xi(X;\boldsymbol{\theta}) = \frac{\partial^2 \log p(X;\boldsymbol{\theta})}{\partial X^2}. \tag{3.20}$$

In practice, the integral mean in (3.19) is replaced by a sample average over the data.

3.4.2 Score-matching estimator for α and β in CNTF algorithm

Using the alpha-beta divergence and based on the assumed distribution in (3.5), the score function for the magnitude STFT coefficients and its derivative will be given by:

$$\psi(X|Z;\alpha,\beta) = -\frac{\partial D_{\alpha\beta}(X\|Z)}{\partial X} = \frac{1}{\beta}X^{\alpha-1}(Z^{\beta}-X^{\beta}), \tag{3.21}$$

$$\xi(X|Z;\alpha,\beta) = -\frac{\partial^2 D_{\alpha\beta}(X\|Z)}{\partial X^2} = \frac{1}{\beta}X^{\alpha-2}\big((\alpha-1)Z^{\beta}-(\alpha+\beta-1)X^{\beta}\big). \tag{3.22}$$

Inserting (3.21) and (3.22) into the SM objective function (3.19), and replacing the integral by a sample mean, we obtain:

$$\mathcal{L}_{SM}(\alpha,\beta) = \sum_{i,k,m}\big[\frac{\alpha+1}{\beta}X^{\alpha}Z^{\beta} - \frac{\alpha+\beta+1}{\beta}X^{\alpha+\beta} + \frac{1}{2\beta^2}X^{2\alpha}(Z^{\beta}-X^{\beta})^2\big]. \qquad (\beta\neq 0) \tag{3.23}$$

Note that in (3.23) we have dropped all indices from $X^{(i)}(k,m)$ and $Z^{(i)}(k,m)$ (i.e., replaced them with X and Z) for simplicity. For the singularity point $\beta = 0$, we extend (3.23) by continuity which yields:

$$\mathcal{L}_{SM}(\alpha,0) = \sum_{i,k,m}\big[\big(-1+(\alpha+1)\log(\frac{Z}{X})\big)X^{\alpha} + \frac{1}{2}\big(X^{\alpha}\log(\frac{Z}{X})\big)^2\big] \tag{3.24}$$

The optimum values for α and β are thus found by searching for a (α,β) pair in the convexity region ($\frac{1-\beta}{\alpha}\in[0,1]$) which minimizes $\mathcal{L}_{SM}(\alpha,\beta)$.

The problem with the above procedure for divergence selection is that it requires the true values of the nonnegative factors (in order to compute $Z^{(i)}(k,m)$), which are not available in practice. We will thus take a two-step approach by alternately optimizing over the factor values and divergence parameters for a certain number of iterations. A summary of the final proposed CNTF dereverberation algorithm is provided in Algorithm 1 at the end of this chapter.

3.5 Experiments

3.5.1 Speech Data

To provide a flexible framework for choosing different microphone configurations and speaker locations, we use room impulse responses from Aachen Impulse Response (AIR) dataset [1]. This is a collection of RIRs recorded in real rooms with different T_{60} values at different source-to-microphone distances. These RIRs have been measured using maximum length sequences of degree 15 as the excitation signal. The measurements have been performed at a sampling rate of 48 kHz with an accuracy of 24 bits, using professional audio equipment providing high-quality and low-noise measurements. More detailed information about the AIR dataset can be found in [1].

To generate reverberant speech data, we convolve TIMIT utterances with different RIRs from the AIR dataset. We use the standard TIMIT data partitions, consisting of 462 speakers for train and 168 speakers for test. This produces a medium-vocabulary ASR task for distant read speech which allows us to concentrate on the evaluation of our front-end enhancement framework.

The AIR dataset provides RIRs from different rooms with different reverberation times. These include an office room ($T_{60} \simeq 430$ ms), a stairway area ($T_{60} \simeq 800$ ms) and a lecture hall ($T_{60} \simeq 800$ ms). For the majority of experiments reported here, we use the RIRs from the stairway area which have been measured at different distances (1m, 2m, 3m) and different azimuth angles for each distance (ranging from 0° to 180° with step sizes of 15°). This allows us to test the algorithm for different source and microphone configurations. The microphone locations used in this study are illustrated in Fig. 3.3, where they have been numbered for easy referencing. The experiments in Section (3.5.4) use RIRs from the office area and lecture hall as well. This will enable us to have diverse reverberation characteristics for experiments with multi-condition training.

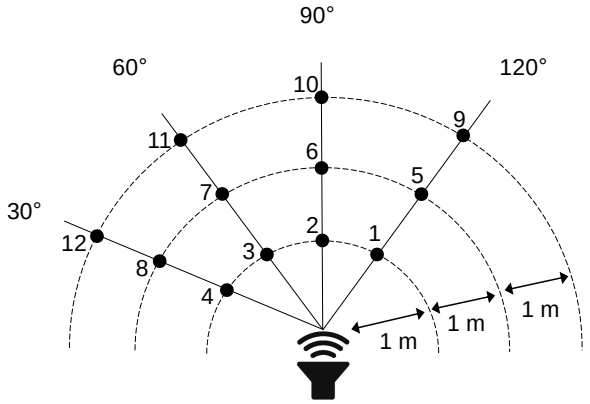


Figure 3.3. Microphone positions used in ASR experiments.

3.5.2 ASR system setup

All ASR experiments reported here use hybrid DNN-HMM acoustic models and a trigram language model created using TIMIT transcriptions (resulting in a medium-vocabulary task with approximately 6000 words). Using a frame size of 25 ms and a skip rate of 10 ms, we extract Mel-filterbank features with 24 Mel filters along with their first and second order derivatives (i.e., delta and double-delta features). We choose Mel-filterbank coefficients because they have been shown to provide consistently better accuracies with DNN-based acoustic models compared to Mel-frequency cepstral coefficients (MFCCs) which are commonly used in GMM-HMM systems [11]. Utterance based mean and variance normalization has been used for these features in all experiments. We initially train tied-state triphone GMM-HMM acoustic models on the clean MFCC features from the training data using the Kaldi speech recognition toolkit [25]. The trained models are then used for forced-alignment of the training data to obtain frame-level senone labels for the training features. Using these labels, we train a deep neural network acoustic model. The inputs to the DNN are concatenated features from a context window of 11 frames. Considering the limited training data in the TIMIT corpus and in order to prevent overfitting, we use a DNN with 3 hidden sigmoid layers each containing 1024 nodes. A softmax output layer in the DNN converts the activations of the last hidden layer into senone posteriors. The DNN parameters are tuned

by stochastic gradient descent (SGD) using error back-propagation to minimize the frame-level cross-entropy between the DNN outputs and the ground-truth senone labels from the forced alignment. The DNN training consists of 30 epochs over the training data with a fixed learning rate of 0.04, using a minibatch size of 256 features. The described clean-trained acoustic model results a word error rate of 3.1% on the clean test set of TIMIT, which is expected for a medium-vocabulary matched train and test experiment.

For recognizing reverberant test data, we first compute magnitude STFTs for all channels using a frame size of 64 ms and a skip rate of 16 ms. These are then jointly processed by 10 iterations of the CNTF dereverberation algorithm to produce an estimate of the clean speech spectrogram. The choice of 10 iterations was experimentally verified to be adequate to provide best ASR accuracy. The CNTF algorithm uses a filter length of $L = 16$ for all $H^{(i)}(k, p)$, and the filters are initialized with $H^{(i)}(k, p) = 1 - p/2L, (p = 0, \cdots, L - 1)$. We perform experiments both with the commonly used Euclidean distance measure (and some other well-known fixed divergences) as well as with $\alpha\beta$-divergence using optimum values for α and β given by the score-matching estimator discussed in Section 3.4. The estimated clean speech magnitude STFT is used together with the phases from one of the channels (first microphone) to reconstruct an estimate for the clean speech waveform, which is finally used for Mel-filterbank feature extraction.

3.5.3 ASR results and algorithm analysis

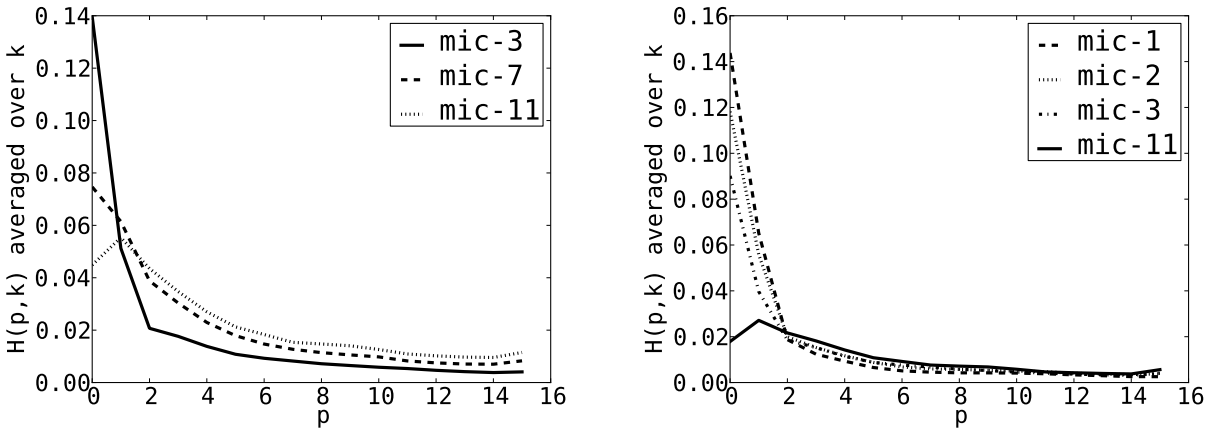
Table 3.2 shows the results of ASR experiments performed to assess the performance of the CNTF algorithm in different DRR scenarios with different microphone configurations. All experiments in this section use subsets of the microphones shown in Figure 3.3. The word error rates shown in the table indicate the effectiveness of the proposed dereverberation strategy in both single-channel and multi-channel scenarios. Using a single-channel version of the algorithm in Table 3.2 (i.e. setting $C = 1$) with the Euclidean distance measure, relative

Table 3.2. Word Error Rates (%) in ASR experiments with clean-trained models*

number of channels	Feature Processing	$d = 2m$ $DRR \approx$ 2.1 dB	$d = 3m$ $DRR \approx$ -0.8 dB
C=1	No enhancement	21.0	29.9
	CNTF ($\alpha = \beta = 1$) (Euclidean Distance)	8.7	13.0
	CNTF ($\alpha = 1, \beta = 0$) (KL Divergence)	8.5	12.8
	CNTF ($\alpha = 1, \beta = -1$) (IS Divergence)	9.4	13.6
	CNTF (SM estimates for α, β)	8.5	12.7
C=2	CNTF ($\alpha = \beta = 1$) (Euclidean Distance)	7.2	10.2
	CNTF ($\alpha = 1, \beta = 0$) (KL Divergence)	6.6	9.8
	CNTF ($\alpha = 1, \beta = -1$) (IS Divergence)	7.0	10.7
	CNTF (SM estimates for α, β)	6.2	9.2
C=4	CNTF ($\alpha = \beta = 1$) (Euclidean Distance)	5.9	8.1
	CNTF ($\alpha = 1, \beta = 0$) (KL Divergence)	5.5	7.6
	CNTF ($\alpha = 1, \beta = -1$) (IS Divergence)	5.9	8.3
	CNTF (SM estimates for α, β)	4.8	6.7

* Single-channel experiments use microphones at 90° (i.e. microphones 6,10) in Fig. 3.3. Dual-channel experiments use microphones at 30° and 90° (i.e. microphones 6,8,10,12).

improvements of +58.5% and +56.5% are provided over the raw filterbank features for source-to-microphone distances of $d = 2m$ and $d = 3m$. The relative WER improvements provided by dual-microphone and four-microphone configurations over the single-channel scenario are +17.2% and +32.2% for $d = 2m$, and +21.5% and +37.7% for $d = 3m$. The improvements provided by adding more microphones is more significant in low-DRR scenarios. It can be observed that although the DNN-based acoustic model shows some inherent robustness to reverberation (indicated by the starting WER of 29.9% in the low-DRR condition of $d = 3m$), the proposed multichannel front-end enhancement strategy can provide significant improvements over the clean-trained DNN baseline. Also shown in Table 3.2 are the WERs obtained with different divergence measures in each scenario. Although all measures are able



(a) 3 microphones at distances of $1m$, $2m$ and $3m$

(b) Unbalanced DRRs: 3 microphones at $d = 1m$, one microphone at $d = 4m$

Figure 3.4. Estimates of RIR spectral envelopes $H^{(i)}(k, p)$ (averaged across all frequencies $k = 0, \cdots, K-1$)

to provide considerable improvements, the best ASR accuracy in most cases is achieved by the (α, β) pair given by the score-matching estimator introduced in Section 3.4.

To better understand the operation of the algorithm, we have plotted the RIR spectral envelopes ($H^{(i)}(k, p)$) discovered by the CNTF algorithm in Figure 3.4 for different configurations. In the first experiment, we use a 3-channel version of the algorithm for three microphones located at distances of 1m, 2m, and 3m from the source (microphones 3, 7, and 11 in Figure 3.3). The resulting RIR envelopes estimated by the CNTF algorithm for each channel are then averaged across all frequencies and plotted in Figure 3.4a. It can be observed that the algorithm has correctly identified the RIR spectral envelope characteristics from the reverberant spectrograms. Microphone 3 at $d = 1m$ has a fairly high DRR which results in a fast exponential decay of the corresponding RIR's spectral envelope. In contrast, the RIR spectral envelope of microphone 11 at $d = 3m$ has a much slower energy decay due to the lower DRR of this channel.

We also perform a second experiment which is designed to illustrate the algorithm's performance when we have highly unbalanced DRRs among the channels. We test a 4-channel example of such situations, in which three channels (microphones 1, 2, and 3) are

fairly close to the source ($d = 1m$), but the fourth channel (microphone 11) is farther away at $d = 3m$. The resulting estimates of $H^{(i)}(k,p)$ given by the CNTF algorithm are averaged across all frequencies and plotted in Figure 3.4b. It is observed that the algorithm has automatically identified the low-DRR channel and has assigned much smaller values to the corresponding $H^{(i)}(k,p)$. Considering the update equation of (3.14), this will in turn reduce the contribution of the low-DRR channel to the final estimate of the clean speech spectrogram. Thus, the algorithm is capable of blindly adjusting the channel contributions according to their signal quality. This is a very useful characteristic in a blind scenario where there is no information about the source and microphone locations.

3.5.4 Experiments with multi-condition and enhanced training data

A DNN acoustic model trained on multi-condition data from various environments is becoming increasingly popular to handle far-field ASR tasks [65]. If such multi-condition data is available, a DNN with a sufficient number of hidden layers usually provides very good results, outperforming most other approaches. In this section, we study the proposed algorithm's performance in situations where a multi-condition training data is available. We compare three different cases. The first method uses clean-trained acoustic models and applies CNTF to reverberant test data in order to reduce the mismatch. The second approach uses reverberant data from multiple different rooms and distances to train multi-condition models, and uses reverberant features directly for decoding (no feature processing). The third approach applies CNTF to both training and test data to further reduce the mismatch.

The experiments in this section use RIRs from all three environments in the AIR dataset (office room, lecture hall, stairway). Using different source-to-microphone distances available for each room, we have a collection of 62 different RIRs to create a multi-condition training data which is diverse in terms of reverberation characteristics. The multi-condition data is created by convolving each utterance in the TIMIT training set with a randomly selected

Table 3.3. Word Error Rates (%) in ASR experiments with clean, multi-condition, and CNTF-enhanced training data*

Train data	Features	Office Room ($T_{60} \simeq 0.43s$) (mic locations: d=2m, d=3m)	Stairway Area ($T_{60} \simeq 0.8s$) (mic locations: d=2m, d=3m)	Lecture Hall ($T_{60} \simeq 0.8s$) (mic locations: d=4m, d=5.5m)
clean	fbank + CNTF (1ch)	8.8	10.8	15.9
clean	fbank + CNTF (2ch)	8.4	9.2	12.9
multi-cond.	fbank (1ch)	6.6	7.3	10.6
multi-cond. (w/ CNTF)	fbank + CNTF (1ch)	6.3	6.6	9.1
multi-cond. (w/ CNTF)	fbank + CNTF (2ch)	5.9	6.3	8.1

* Single channel experiments use the microphone closer to the source.

RIR from this pool. The resulting reverberant signal is time-synchronized with the original clean utterance so that the ground-truth senone alignments from the clean data can be used for training. We consider three different test environments. The first set of experiments are in an office room ($T_{60} \simeq 0.43s$) with two microphones placed at $d = 2m$ and $d = 3m$ from the source. The second set of experiments are conducted in the stairway area which possesses a longer reverberation time ($T_{60} \simeq 0.8s$), keeping the same distances of $d = 2m$ and $d = 3m$. For the last set of experiments, we use a lecture hall which has a similar reverberation time of $T_{60} \simeq 0.8s$, but we increase source-to-microphone distances to $d = 4m$ and $d = 5.5m$. For all of the experiments in this section, fixed values of $\alpha = 1$ and $\beta = 1$ have been used for CNTF updates.

The results of ASR experiments in the above three cases are shown in Table 3.3. All multichannel experiments here have unbalanced DRRs, i.e., one microphone is located farther from the source compared to the other (note the distances mentioned in Table 3.3). An important observation from the table is that although the farther microphone has a

more corrupted signal (lower DRR), including it in a joint CNTF processing with the closer channel is still superior to a channel selection strategy in which only the closer microphone is identified and used [60]. In this case, based on the discussion in Section 3.5.3, although the reverberation filters $H^{(i)}(k,p)$ place a smaller weight on the low-DRR channel, they are still able to draw useful information from this channel which helps ASR accuracy. Another result to be noted in Table 3.3 is that although the multi-condition DNN baseline (third row) provides considerable robustness to reverberation and good accuracy, the error rates can be further reduced by applying CNTF processing to both train and test data, because this will further reduce the mismatch. Therefore, while CNTF front-end processing is necessary in mismatched conditions with clean-trained models, it can also be beneficial for multi-condition models. We believe that for very large datasets spanning very diverse environmental conditions, multi-condition trained DNN acoustic models are sufficient without front-end processing. However, for moderate data sizes which inevitably exclude many possible reverberation conditions, reducing the mismatch by applying CNTF processing will prove helpful, as verified by the results in Table 3.3.

3.6 Summary

We presented an algorithm for reverberation-robust distant speech recognition using distributed far-field microphones based on convolutive nonnegative tensor factorization (CNTF). The developed algorithm attempts to remove the convolutional effects of the RIR from the received signals. In the single channel case, the algorithm simplifies to nonnegative matrix factorization and attempts to decompose each subband envelope into a convolution of two components, one being the clean speech subband envelope and the other representing the convolutive effects of the RIR. In the multichannel case, the algorithm makes use of the additional observations to improve this decomposition. In each subband, the algorithm identifies a common component among the subband envelopes of the different channels (which is the

clean speech component), and discards the convolutive differences between the channels as RIR distortions. The proposed CNTF algorithm explicitly addresses the reverberation tail effect by attempting to remove the long-term correlations that are introduced in the sub-band envelopes by the RIR. In addition to the clean-trained scenarios, it was shown that the proposed algorithm can also be helpful with multi-condition training data, since applying CNTF to both train and test signals will help reduce the mismatch.

Algorithm 1: CNTF dereverberation

`// Superscript * represents complex conjugate.`
`// All variables with a bar represent FFT domain variables.`

Input: $X^{(i)}(k,m) \quad (\forall i,k,m)$

$m \in \{0,\cdots,M-1\}, k \in \{0,\cdots,K-1\}, i \in \{1,\cdots,C\}, p \in \{0,\cdots,L-1\}$

Set $\alpha = 1, \beta = 1$

Set $F = M + L - 1$

repeat

Initialize $H^{(i)}(k,p) = 1 - \frac{p}{2L} \quad (\forall i,k,p)$

Initialize $S(k,m) = X^{(1)}(k,m) \quad (\forall k,m)$

for *iter* $= 1$ *to* N **do**

`// Computing` $Z^{(i)}(k,m)$ `via FFT multiplication:`

$\bar{H}^{(i)}(k,f) = \sum_{p=0}^{L-1} H^{(i)}(k,p) \exp(-j\frac{2\pi pf}{F})$

$\bar{S}(k,f) = \sum_{m=0}^{M-1} S(k,m) \exp(-j\frac{2\pi mf}{F})$

$Z^{(i)}(k,m) = \sum_{f=0}^{F-1} \bar{H}^{(i)}(k,f)\bar{S}(k,f) \exp(j\frac{2\pi mf}{F})$

`// Intermediate variables:`

$Y^{(i)}(k,m) = X^{(i)}(k,m)^{\alpha} Z^{(i)}(k,m)^{\beta-1}$

$V^{(i)}(k,m) = Z^{(i)}(k,m)^{\alpha+\beta-1}$

$\bar{Y}^{(i)}(k,f) = \sum_{m=0}^{M-1} Y^{(i)}(k,m) \exp(-j\frac{2\pi mf}{F})$

$\bar{V}^{(i)}(k,f) = \sum_{m=0}^{M-1} V^{(i)}(k,m) \exp(-j\frac{2\pi mf}{F})$

`// Compute correlations via FFT multiplication:`

$C_{YS}^{(i)}(k,p) = \sum_{f=0}^{F-1} \bar{Y}^{(i)}(k,f)\bar{S}^{*}(k,f) \exp j\frac{2\pi pf}{F}$

$C_{VS}^{(i)}(k,p) = \sum_{f=0}^{F-1} \bar{V}^{(i)}(k,f)\bar{S}^{*}(k,f) \exp j\frac{2\pi pf}{F}$

$C_{YH}^{(i)}(k,m) = \sum_{f=0}^{F-1} \bar{Y}^{(i)}(k,f)\bar{H}^{(i)^{*}}(k,f) \exp j\frac{2\pi mf}{F}$

$C_{VH}^{(i)}(k,m) = \sum_{f=0}^{F-1} \bar{V}^{(i)}(k,f)\bar{H}^{(i)^{*}}(k,f) \exp j\frac{2\pi mf}{F}$

`// Update nonnegative factors:`

$H^{(i)}(k,p) \leftarrow H^{(i)}(k,p)\frac{C_{YS}^{(i)}(k,p)}{C_{VS}^{(i)}(k,p)}$

$S(k,m) \leftarrow S(k,m)\frac{\sum_{i=1}^{C} C_{YH}^{(i)}(k,m)}{\sum_{i=1}^{C} C_{VH}^{(i)}(k,m)}$

`// Normalization`

$H^{(i)}(k,p) \leftarrow \frac{H^{(i)}(k,p)}{\sum_{i,p} H^{(i)}(k,p)}$

end

`// Update` α `and` β`:`

$\mathcal{L}_{SM}(\alpha,\beta) = \sum_{i,k,m} \left[\frac{\alpha+1}{\beta} X^{\alpha} Z^{\beta} - \frac{\alpha+\beta+1}{\beta} X^{\alpha+\beta} + \frac{1}{2\beta^2} X^{2\alpha}(Z^{\beta} - X^{\beta})^2\right]$

$(\alpha,\beta) = \arg\min_{(\alpha,\beta)} \mathcal{L}_{SM}(\alpha,\beta) \qquad \frac{1-\beta}{\alpha} \in [0,1]$

until α *and* β *converge, or maximum iterations is reached*

return $S(k,m) \quad (\forall k,m)$

CHAPTER 4

MULTI-DOMAIN ADVERSARIAL TRAINING OF NEURAL NETWORK ACOUSTIC MODELS FOR IMPROVED FAR-FIELD ROBUSTNESS

Our discussion on far-field robustness in chapter 3 was focused on multi-channel front-ends to compensate the far-field effects and reduce the feature distortions. Beginning in this chapter, we take a different approach to robustness by focusing on back-end (acoustic model) robustness in single-channel scenarios. This chapter presents a novel strategy for training neural network acoustic models based on adversarial training which improves robustness to recording conditions[1]. We provide a motivating study on the mechanism by which a deep network learns environmental invariance, and discuss some relations with existing approaches for improving the robustness of DNN models.

4.1 Robust representation learning with multi-condition DNNs

A major advantage of DNNs to conventional models is their representation learning capability which facilitates the use of simple raw features with less task-specific feature engineering. As a consequence, the adoption of DNN-based acoustic models has brought a shift of focus in strategies to address the problem of distant (far-field) speech recognition. Traditionally, array processing or single-channel enhancement have been considered as the primary solutions. In contrast, the use of deep learning in ASR has popularized an alternative solution which addresses the far-field problem from an acoustic modeling perspective, directly using reverberant features without any manually-designed enhancement pipelines. By training models on a large corpus of far-field speech from different rooms with different acoustic properties, the network learns to perform the necessary feature transformations within the hidden layers

[1]This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice.

to compensate for far-field distortions, resulting in robust final-layer representations that are less impacted by recording conditions. Assuming there is sufficient diversity in the train data in terms of reverberation times, direct-to-reverberation ratios, etc., the resulting model is expected to generalize to unseen recording conditions.

Modeling the feature distributions of speech phonemes based on reverberant data is a difficult learning problem due to the long-term context-dependent distortions introduced by reverberation which span multiple frames. However, the remarkable capability of RNNs to remember relevant past information enables them to sufficiently model the long-term correlations in reverberant speech. This makes RNN-based acoustic models very attractive for multi-condition training on reverberant data. The resulting model often outperforms most front-end engineered solutions which aim to directly compensate distortions by manual feature enhancement. For moderate data sizes, feature enhancement may help when employed in a noise-adaptive manner [66] (i.e., applied to both train and test utterances). However, as more training data is added, the impact from such enhancement techniques is reduced. As a result, the current focus in far-field ASR is on more effective DNN/RNN architectures and improved representation learning algorithms that can automatically derive useful representations directly based on far-field data.

The studies in [67] and [68] introduce modifications to the standard Long Short Term Memory (LSTM) networks in order to improve information flow in time and through the network layers. They report improvements in ASR accuracy by using these alternative recurrent structures. Other studies attempt to improve the model by extracting auxiliary features which describe the recording condition and appending them to the spectral features [69–71]. These *factor-aware* approaches are expected to guide the training procedure by providing additional information about the recording environment. Another group of approaches make use of parallel clean data to improve training. They show that if a training dataset consisting of pairs of close-talk and far-field utterances is available, the model training can be guided to jointly learn both enhancement and recognition [70, 72, 73].

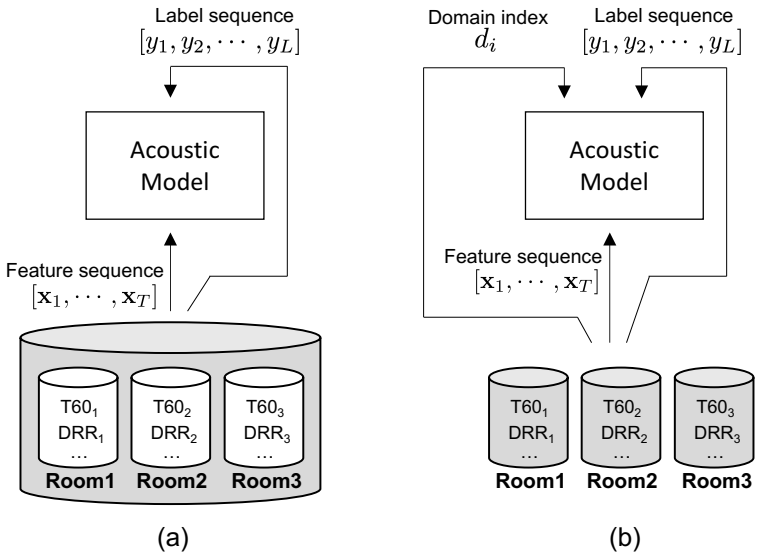


Figure 4.1. (a) Conventional multi-condition training: far-field data from different rooms is combined into a single train set (b) Multi-domain adversarial training: room labels are used during training to achieve improved invariance to recording conditions.

4.2 Multi-domain training

The current common way to build multi-conidition acoustic models is to compile speech data from different recording environments (e.g., different rooms with different reverberation times) into a single train set, and then build models based on this combined dataset, ignoring the environment labels during training (Figure 4.1(a)). This is similar to the early work on multi-style training for robust speech recognition under stress [74, 75]. In the far-field scenario, by providing many training examples from different environments, the network is expected to derive robust representations that are invariant with respect to the range of recording conditions. Ideally, the recording environment of an utterance should be indistinguishable from the features of the last hidden layer. However, as will be shown in Section 4.5.1, in practice there is sustained residual information concerning the recording environment at the last hidden layer.

The goal of this chapter is to improve multi-condition training of RNN acoustic models by incorporating knowledge about the recording environment of each utterance into the training

process. We assume a dataset of far-field speech from multiple different rooms is available, in which each utterance is labeled not only by a phoneme or character sequence, but also by an additional label that indicates which room it belongs to. The intention is to use this available corresponding meta-data about the training utterances in order to force better invariance with respect to recording conditions (Figure 4.1(b)). To achieve this, we propose to use adversarial training with respect to a *domain recognizer* that is built on top of the hidden layers in the nework, and is expected to predict the environment label of an input utterance. We refer to the different rooms and recording conditions as multiple *domains* within the train data which exhibit slightly different distributions. In this context, we use *multi-domain* training in place of multi-condition training to emphasize the use of domain labels during training.

The idea of adversarial training was first proposed in [76] as a generative model that could learn the data distribution by explicitly trying to make samples from the modeled distribution indistinguishable from actual training examples. Generative Adversarial Networks (GANs) were used in [76] for the task of realistic image generation. Since then, GANs have been successfully used in many different tasks and applications where invariance is needed with respect to different distributions. For example, the works in [77] and [78] use adversarial training for effective transfer learning by making use of an unlabeled adaptation set from a target distribution. Also, the study in [79] uses a Gradient Reversal Layer (GRL) similar to [78] in order to improve noise robustness of hybrid DNN-HMM models by negating the gradients derived from a noise-type classifier.

To motivate the use of GANs for achieving far-field robustness in ASR, we first present an analytic study on how RNN acoustic models automatically learn to compensate environment variabilities from multi-condition data. This study reveals an opportunity to employ adversarial training in order to enforce better environment invariance at the intermediate layers.

4.3 Environment invariance in hidden representations

4.3.1 Environment-specific features in hidden layers

When trained on a large dataset of far-field speech from different environments with diverse acoustic properties, a DNN-based acoustic model often provides competetive results without requiring manually-designed dereverberation or feature enhancement strategies. This is because the internal representations of a deep network become increasingly invariant to those varations in data which are irrelevant to the classification task. For an RNN acoustic model trained on multi-condition far-field data, we expect the hidden features to be less sensitive to the recording environment as we move toward the deeper layers in the network. At the final hidden layer, the discovered representations should be maximally discriminant with respect to speech phonemes, with minimum variance resulting from the recording environment.

Figure 4.2 shows the hidden representations from a 3-layer RNN-CTC model that are mapped into a 2D plane through a Linear Discriminant Analysis (LDA) projection. The network has been trained on far-field data from the AMI corpus [6], which contains speech recorded by table-top microphones in 3 different meeting rooms (more details about the AMI data is provided in Section 4.5.1). The input features from the different rooms are significantly overlapped in the input feature space (Figure 4.2(a)). This is expected because there is no explicit room-specific feature in the input feature vectors (only Mel filterbank coefficients which represent spectral characteristics of speech). However, it can be observed in Figure 4.2(b) that in the first hidden layer, the network tries to map the features from the different rooms (domains) into separate subspaces. In other words, the first hidden transformation automatically learns to extract features that are indicative of the recording environment of an utterance. Note that this is achieved while the network receives no supervising information about the environment to which each training example belongs. The only supervision provided to the network during training is the output character sequence.

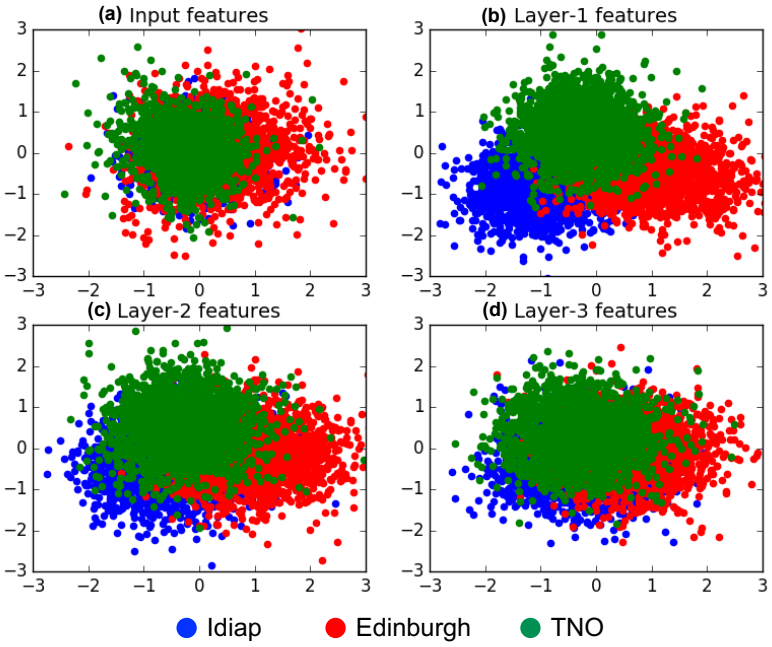


Figure 4.2. RNN hidden features projected onto the 2D plane (best viewed in color). Reprinted with permission, from S. Misamadi, J. H. L. Hansen, On multi-domain training and adaptation of end-to-end RNN acoustic models for distant speech recognition, Interspeech 2017.

The network implicitly learns that in order to accurately predict the label sequence, it first needs to project the data from different domains to different subspaces by extracting environment-specific features. The subsequent layers, however, show an increasing overlap between the data from different rooms, indicating the network's attempt to dervie room-invariant features. The process of representation learning in multi-condition DNNs can thus be viewed as a two-step procedure. The initial layers map data from different domains into separate subspaces by extracting room-specific features, and the subsequent layers learn to use this encoded domain knowledge to compensate the differences.

4.3.2 domain classification accuracy in hidden layers

To quantitatively assess this propagation of domain information within the network, we employ a simple linear classifier based on features from the different layers to predict which

of the three meeting rooms of the AMI corpus an input utterance belongs to. Here, we explicitly use the room labels to train a logistic regression model as a room classifier while keeping the parameters of the RNN-CTC model fixed (See Section X for details about the setup). The resulting accuracies are depicted in Figure 4.3. The solid lines indicate the accuracy of room classification based on the hidden features of different layers, when the model is trained on far-field data. It is observed that predicting the recording rooms directly based on input filterbank features using a linear classifier results in low accuracy, since there is no feature in the input which explicitly describes room characteristics. However, using the first hidden layer representations results in a much higher accuracy for both 3-layer and 6-layer RNN models. This indicates that the network has automatically derived features in the first layer that bear information about the characteristics of the recording environment. The subsequent layers, however, show gradually decreasing domain classification accuracy, which indicates that the network is trying to discover domain-invariant features as we move towards the output layer. Note that these results have been obtained using a simple linear model for domain classification (logistic regression). By using approporiate nonlinear feature transformations (i.e., an arbitrarily deep domain classifier), we can achieve higher accuracies even with input filterbank features. However, the goal here is to simply measure the amount of domain knowledge already encoded in the feature representations at each layer, without any supervised nonlinear transformations to extract those features.

The domain classification accuracy at the first hidden layer of the deeper (6-layer) network is superior to the 3-layer network. This means that a deeper network allows for better extraction of environment-specific information in the initial layers. Second, the domain accuracy at the last hidden layer of the deeper network is lower, which demonstrates the ability of the deeper network to better achieve domain-invariant final representations. However, even at the final layer of the deeper network, domain classification accuracy is still significantly higher than chance-level, indicating the presence of some residual domain-specific information at

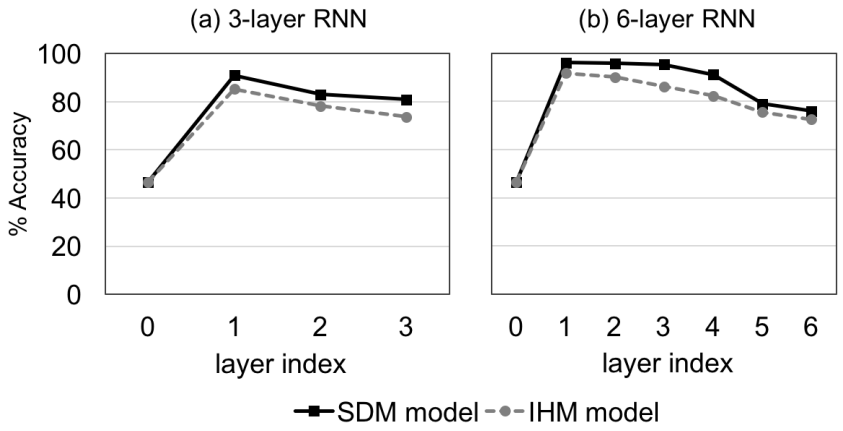


Figure 4.3. Room classficiation accuracies based on features from different hidden layers in a 3-layer and a 6-layer RNN trained on speech data from AMI corpus.

this stage. In other words, although the supervison provided by label sequences encourages the network to minimize the influence of room-related factors in the final representations, in practice it cannot achieve complete domain invariance. The proposed adversarial training framework described in Section 4.4 uses room labels during training to better enforce this desired invariance in the hidden representations.

It is worth noting that the different rooms in the AMI corpus differ not only in terms of the acoustic properties of the rooms, but also in terms of the speakers in each room. Therefore, by training on the Single Distant Microphone (SDM) channel, the model learns differences that result from both environment and speaker characteristics. Based on the hidden features of the SDM model, it is not clear how much of the encoded domain knowledge pertains to room characteristics versus speaker differences. To quantify the role of each factor, Figure 4.3 also shows the domain accuracies with a clean-trained model (dashed lines) which is trained on data from the Individual Headset Microphones (IHM). The input features in this case are still far-field SDM features, but they are passed through the IHM-trained model to produce hidden features. The IHM model uses close-talking signals and thus cannot learn any specific information concerning room characteristics. Thus, any domain discrimination in this case

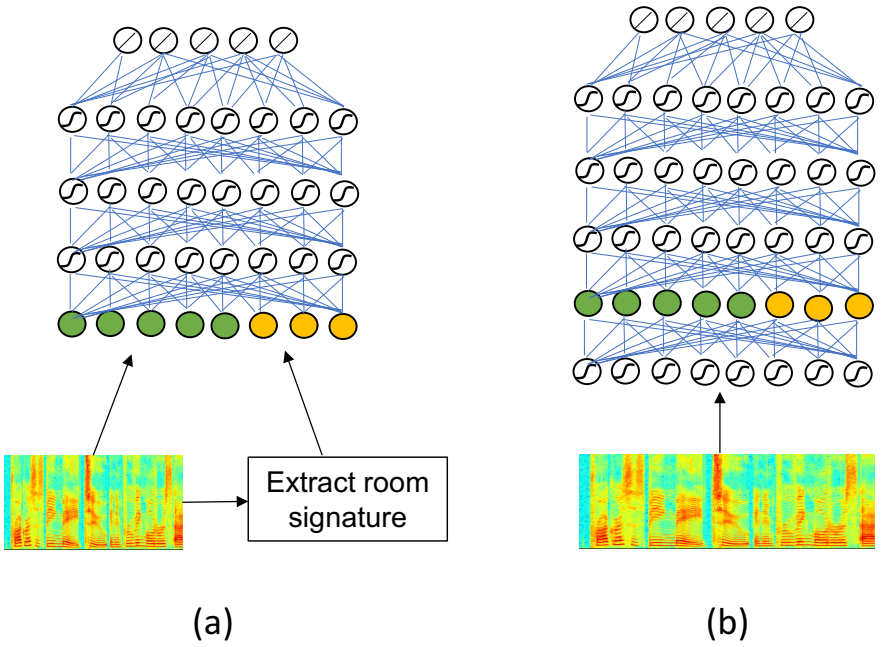


Figure 4.4. (a) factor-aware training where manually extracted room features are appended to input feature vectors, (b) Given sufficient far-field training data, the network automatically learns to extract room features in the hidden representations. (best viewed in color)

is solely due to speaker differences. The SDM model, in contrast, learns both speaker and environment differences, and thus results in higher domain classification accuracy. Based on this consideration, the difference between solid and dashed lines in Figure 4.3 indicates the amount of added domain information due to room differences.

4.3.3 Links to factor-aware training

A group of approaches collectively referred to as *factor-aware* training [69–71] attempt to improve neural network acoustic models by extracting manually designed features that are indicative of a variability factor in the signal (speaker, room, etc.) and appending them to spectral features for ASR. By having access to this extra information about the signal, the network is able to derive more robust features that compensate those variations. In the context of far-field ASR, the auxilliary feature may descibe distance, reverberation time,

DRR, spectral envelope of RIR, etc. The analysis provided in the previous section indicates that by having access to sufficient far-field speech from each room, the network is able to automatically derive room-specific features in the initial hidden layers, without relying on a separate module. In essence, factor-aware training uses input features that lie on separate subspaces for each room. However, given sufficient depth, a multi-condition model trained on far-field data tries to automatically achieve a similar mapping into separate subspaces in the initial layers. Figure 4.4 compares the flow of domain-specific knowledge in factor-aware versus multi-condition training.

4.4 Multi-domain adversarial acoustic model training

4.4.1 Generative Adversarial Networks (GAN)

The basic idea in GANs is to set up a game between two learners, which are referred to as generator and discriminator. The generator's task is to create samples from a distribution that closely resembles the training data. The role of the discriminator is to distinguish between real and fake samples, (i.e., to recognize whether a sample is an actual training example or created by the generator). The idea of GANs was originally proposed in [76] and used for the task of generating images, where the generator is expected to map from random noise to a realistic image. In the context of neural networks, adversarial training refers to optimizing the parameters from two parts of a network with opposing objective functions. In the image generation task, the discriminator is a binary neural network classifier which is trained to maximize the probability of correct decision on whether its input is supplied from the training set or provided by the generator. The generator is another network which is trained to fool the discriminator by maximizing the probability that its output representation is recognized as a real image by the discriminator. In other words, the generator is trained to yield an incorrect prediction in terms of real or fake images. As a result of this competition

between the two networks, the generator will try to produce images that very closely resemble actual images in the training data, therefore making it difficult for the discriminator to distinguish real images from the generator outputs. GANs have gained rapid popularity as state-of-the-art in generative modeling in many different tasks and fields [80, 81]. The idea of adversarial training was used in [77] and [78] for unsupervised adaptation, where instead of real and fake images, the discriminator is trained to distinguish between source-domain data coming from a labeled train set and target-domain data from an unlabled adaptation set. In [78], adversarial training of the generator is implemented by using a gradient reversal layer, which is an identity transform in the forward pass, but negates the gradients during back-propagation. The gradient reversal technique was also used in [79] for a noise-type classifier to improve noise robustness of a hybrid DNN-HMM model.

4.4.2 Multi-domain Adversarial training of RNN-CTC models

Figure 4.5 shows our proposed network architecture for improved training of RNN-CTC acoustic models using multi-domain far-field speech data. The left path in Figure 4.5 (G and W networks) is a standard RNN-CTC model as described in Section 2.3.3. It uses multiple bi-directional recurrent layers on top of the input filterbank features, followed by a softmax layer to yield posterior probabilities for each of the symbols in the character (or phoneme) set. In addition to this main path, there is a secondary network (D) branching out from one of the hidden layers. Here, D is a domain classifier, (i.e., it is expected to recognize the specific recording environment for the input utterance). The domain classifier consists of a few initial dense layers at the frame level (with paramters shared in time), which are expected to map from the hidden representations of the CTC network to a new space with features that are discriminant with respect to the recording environment. Since we need a single decision for the entire utterance, intermediate features from these frame-level layers are aggregated into an utterance-level representation via mean-pooling over all frames. Note

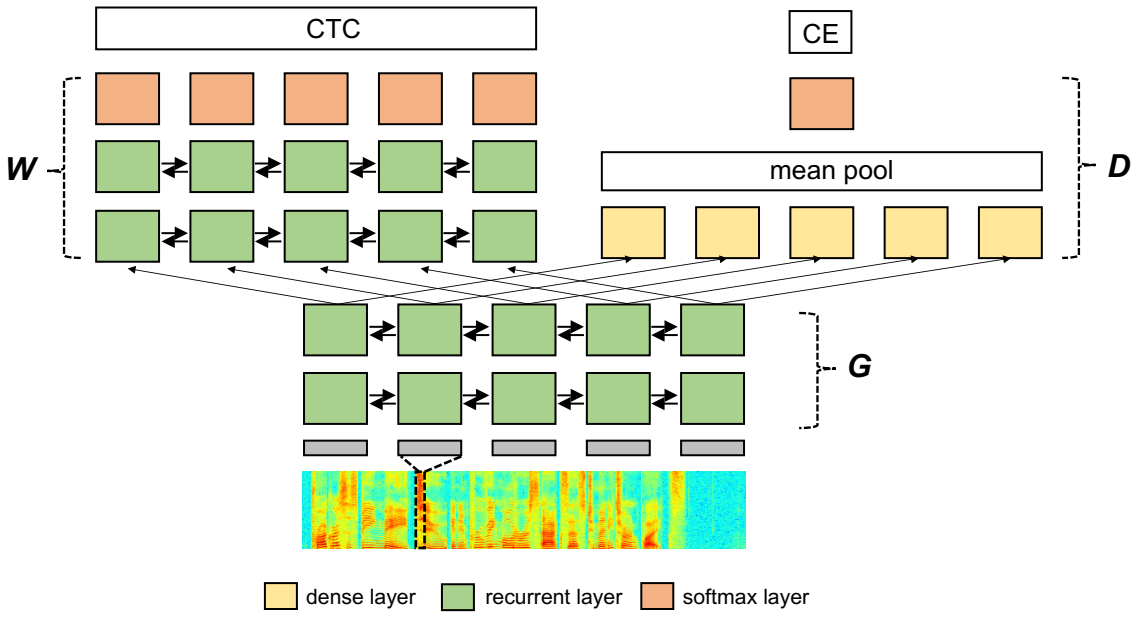


Figure 4.5. Network structure for the proposed multi-domain adversarial training approach.

that there is in general alternate possible ways to train such networks that are expected to map from a sequence of features to a single label. These include frame-wise training where the overall label is assigned to each and every frame (and the resulting errors back-propagated from every frame), as well as different forms of pooling in time (mean, max , final-frame, etc.). For the proposed multi-domain training approach, we have empirically verified that a simple mean pooling of the intermediate domain features over all frames yields the best results. This is in agreement with what has been observed for other sequence classification tasks for speech [82] and video [83].

Assuming that the domain classifier is based on features from the l_G'th recurrent layer, frame-level transformations for domain classification can be written as:

$$\mathbf{v}_t^{(k)} = \sigma\big(\mathbf{R}^{(k)}\mathbf{v}_t^{(k-1)} + \mathbf{c}^{(k)}\big), \tag{4.1}$$

where $\mathbf{R}^{(k)}$ and $\mathbf{c}^{(k)}$ are layer weights and biases, $\sigma(\cdot)$ represents the layer non-linearity, and $\mathbf{v}_t^{(0)} = \mathbf{h}_t^{(l_G)}$. The output layer of the domain classifier is a softmax layer which maps from

the time-averaged domain representations to posterior probabilities for each of the domains:

$$q_i = p(d_i|\mathbf{X}) = \frac{\exp\left(\mathbf{u}_i^T \mathbf{v}\right)}{\sum_{j=1}^{N_D} \exp\left(\mathbf{u}_j^T \mathbf{v}\right)}, \tag{4.2}$$

where,

$$\mathbf{v} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{v}_t^{L_D}. \tag{4.3}$$

Here, d_i denotes the i'th recording environment (domain), N_D is the total number of domains in the train data, and L_D is the number of frame-level layers in the domain classifier.

The parameters in the second part of the CTC network ($\boldsymbol{\theta}_W$) are trained using the CTC cost in the usual way as described in Section 2.3.3. The parameters of the domain classifier ($\boldsymbol{\theta}_D$) are trained to maximize the probability of correct domain prediction, (i.e., by using the cross-entropy cost between the softmax outputs and ground-truth domain labels):

$$J_D = -\sum_{i=1}^{N_D} z_i \log(q_i), \tag{4.4}$$

where $[z_0, \cdots, z_{N_D}]$ is a one-hot encoding of the ground-truth domain label (i.e., z_i is 1 if it corresponds to the correct domain and 0 otherwise). The parameters in the shared section of the network between the CTC task and the domain classifier (i.e., $\boldsymbol{\theta}_G$) are optimized to reduce both the CTC cost and an adversarial domain objective (J_A) which is designed to oppose a correct domain classification:

$$J_G = J_{CTC} + \lambda J_A. \tag{4.5}$$

There are different possible choices for the adversarial cost J_A which are listed in Table 4.1. The simplest approach is to choose $J_A = -J_D$, which corresponds to a minimax game where the domain classifier tries to maximize the probability of the correct domain, while the generator tries to minimize it. Although the operation of GANs is often described using this objective, in practice it suffers from an early saturation of the cost when the domain classifier is providing a correct prediction, thus receiving vanishing gradients and failing to

Table 4.1. Different parameters of a multi-domain RNN-CTC network and their associated costs

parameters	**associated cost**
$\boldsymbol{\theta}_W$	$J_{CTC} = -\log\Big(\sum_{a\in\mathcal{A}}\prod_{t=1}^{T} p\big(s_{a[t],t}\vert\mathbf{X}\big)\Big)$
$\boldsymbol{\theta}_D$	$J_D = -\sum_{i=1}^{N_D} z_i \log(q_i)$
$\boldsymbol{\theta}_G$	$J_G = J_{CTC} + \lambda \sum_{i=1}^{N_D} z_i \log(q_i)$
	$J_G = J_{CTC} - \lambda \sum_{i=1}^{N_D} (1 - z_i) \log(q_i)$
	$J_G = J_{CTC} - \lambda \sum_{i=1}^{N_D} \frac{1}{N_D} \log(q_i)$

achieve the adversarial task [84]. An alternative cost to overcome this problem is to reverse the domain labels for training the generator (i.e., assigning 0 to the correct domain and 1 to the other domains):

$$J_A = -\sum_{i=1}^{N_D} (1 - z_i) \log(q_i). \tag{4.6}$$

Although this works well for the original GAN structure which uses a binary discriminator, for our multi-domain training framework, switching the labels does not lead to a well-defined objective, as it yields multiple *correct* domains and a single incorrect domain. To be applicable to a multi-domain scenario, we propose to use the Kullback-Liebler (KL) divergence between the discriminator outputs and a uniform distribution ($z_i = \frac{1}{N_D}, i = 1, \cdots, N_D$) as the training cost for generator parameters:

$$J_A = -\sum_{i=1}^{N_D} \frac{1}{N_D} \log(q_i). \tag{4.7}$$

Here, instead of using the hard domain labels, the domain posteriors are being compared with a set of soft targets which represent a uniform distribution over all domains. The generator is thus trained to achieve a state of maximum confusion where equal probabilities are assigned to different domains.

Many original GAN studies choose to simultanously optimize all network parameters together according to the objectives in Table 4.1. This means that each mini-batch of data results in an update of $\boldsymbol{\theta}_D$ in the direction of minimizing domain cost and an update of $\boldsymbol{\theta}_G$ to minimize the adversarial cost. Although this is an approximation for the complete iterative training (where we alternate between discriminator and generator updates, keeping one fixed and updating the other), it has been shown to perform well in practice in image generation tasks with feed forward nets. However, to have a stable learning in our multi-domain RNN training, using an iterative optimization strategy was found to be necessary. The resulting train procedure is summarized in Algorithm 2. Each epoch of the algorithm consists of one CTC pass over the train data, followed by domain discriminator updates and generator updates. Note that Algorithm 2 uses Stoachastic Gradient Descent (SGD) updates for simplicity. In practice, adaptive learning rate methods such as RMSprop [85] can be used for faster convergence.

4.4.3 Comparison with multi-task learning

The proposed network architecture in Fig. 4.5 is structurally similar to Multi-Task Learning (MTL) networks where two different classification tasks are solved based on a shared intermediate representation. However, in spite of this structural similarity, multi-domain adversarial training is actually the exact opposite of MTL. This comparison is illustrated in Fig. 4.6. In MTL, we have two different but related tasks to be learned from a single domain of data. The network parameters are optimized to reduce classification error for both tasks. By sharing some of the hidden transformations, the two tasks are expected to support each other. In other words, the intermediate features at the output of the shared section of the network are expected to be discriminant with respect to both tasks. In contrast, in multi-domain adversarial training, we have a single main task which is common to multiple data domains, and we need an intermediate representation which is discriminant for the main classification task but invariant with respect to the domains.

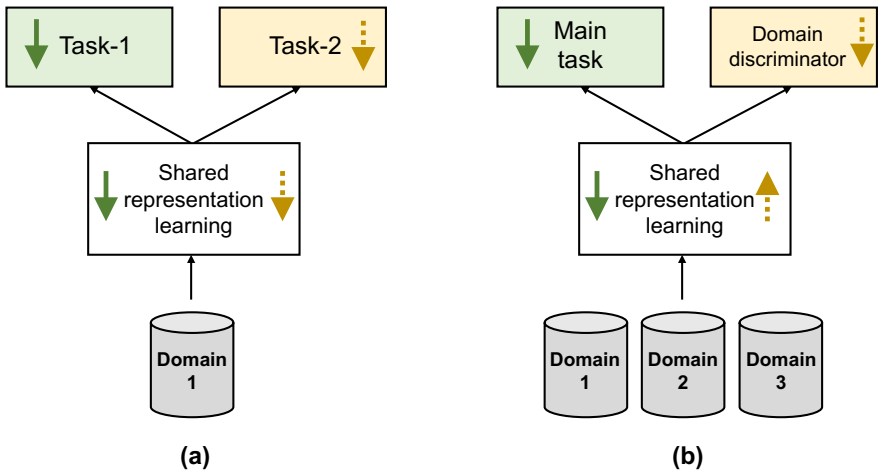


Figure 4.6. Comparison between standard multi-task learning and multi-domain adversarial training. The arrow directions indicate the changes in each cost value resulting from parameter updates. (best viewed in color)

4.5 Experiments

4.5.1 System setup and data

We evaluate the proposed multi-domain training approach on an end-to-end ASR task based on the AMI meeting corpus [6]. The AMI corpus consists of speech recorded in three different meeting rooms[1], using both Individual Headset Microphones (IHM) and a micorphone array placed on the meeting table. We use only a Single Distant Microphone (SDM) from the array to provide far-field speech data for train and test. The SDM channel poses two different problems for ASR: simultanous speech and far-field distortions. To focus on the latter, we remove any utterances that contain overlapped speech regions from both train and test data, which leaves us with approximately 30 hours of train, 4 hours of development, and 4 hours of test data (we use the recommended data partitions for ASR outlined in [86]).

The input features are 24-dimensional Mel filterbank coefficients with speaker-level mean and variance normalization. These are extracted using 25 msec windows at a rate of 100

[1]Three different data collection sites: University of Edinburgh (U.K.), IDIAP research institute (Switzerland), and the TNO Human Factors Research Institute (The Netherlands)

frames per second. We use frame-skipping [87] with a context window of 3 frames to reduce the required computations for RNN training[1].

The baseline acoustic model is a 6-layer recurrent network with bi-directional LSTM (BLSTM) layers containing 128 cells per direction, leading to 256-dimensional hidden layer representations. We choose an output space similar to [88], by having multi-character units (such as *ll* in *tall* and *'s* in *let's*), and by using uppercase characters to indicate the beginning of a word. This makes an output space of size 79, consisiting of 78 character labels plus the blank symbol[2]. Network parameters are optimized using RMSprop [85] with an initial learning rate of 0.001 and a minibatch size of 20 utterances. Training epochs are stopped when no further improvement in Character Error Rate (CER) is observed on the development set. The results are reported using both a simple best-path decode strategy (choosing the most active output at each frame followed by removing blanks and repetitions), and also by a beam-search algorithm with a beam width of 10 to track multiple candidate sequences. In both cases, decoding is based only on acoustic scores from the RNN using no additional lexicon or language information.

Table 4.2 shows the performance of this baseline CTC network when trained on clean speech (IHM channel) and far-field speech (SDM channel). As expected, the clean-trained model results in a high error rate on the SDM test data due to the significant underlying mismatch in this case. The multi-condition model uses far-field (SDM) train data from multiple rooms and speaker-to-microphone distances within the AMI corpus, and thus provides significantly improved far-field robustness, yielding a +23% relative improvement with respect to the clean-trained model.

[1]This has been been shown to have minimal performance effects on CTC models particularly with large datasets [87].

[2]We have empirically verified that using this output space provides consistently better results compared to the alternative approach which uses dedicated space and apostrophe characters [89].

Table 4.2. Baseline Character Error Rates with clean-trained (IHM) and far-field (SDM) models

Train Data	Test Data	CER (best-path)	CER (beam-search)
IHM	IHM	32.8	32.2
IHM	SDM	63.6	62.9
SDM	SDM	49.1	48.5

4.5.2 Results on multi-domain adversarial training

Table 4.3 compares the performances provided by the proposed multi-domain adversarial framework when the domain discriminator branches out from different hidden layers of the network. The domain discriminator is chosen to be a 2-layer network with a frame-level sigmoid layer of 256 nodes, followed by a mean-pool operation across all frames of the utterance and a final softmax layer to provide probabilities of belonging to each of the three AMI meeting rooms. The network is trained according to Algorithm 2, with $\lambda = 20.0$ and $r_D = r_G = 4$ (i.e., each epoch consists of one CTC pass to tune $\boldsymbol{\theta}_W$ and $\boldsymbol{\theta}_G$, followed by 4 iterations of cross-entropy updates on domain discriminator ($\boldsymbol{\theta}_D$) and 4 iterations of adversarial updates on $\boldsymbol{\theta}_G$). As observed in Table 4.3, the best results are obtained when the domain discriminator is based on layer-2 features, which yields a +3.3% relative improvement over a multi-condition baseline that does not use domain labels during training. The overall improvement with respect to the clean-trained baseline is +25.4% in this case.

Several observations can be made based on the results in Table 4.3. First, note that when the domain discriminator is based on features from the first hidden layer, the resulting performance is worse than the baseline. In other words, forcing environment invariance from the very first layer actually harms performance. This can be understood by considering the learning mechanism described in Section 4.3. In the first hidden layer, the network extracts domain-specific features that are indicative of the recording conditions. The subsequent layers use this encoded domain knowledge to derive the robust final representations. Therefore,

Table 4.3. Character Error Rates provided by multi-domain adversarial training, when the domain discriminator is based on the features from different hidden layers.

Discriminator branch layer	CER (best-path)	CER (beam-search)
None (Baseline)	49.1	48.5
Layer-1	53.1	52.9
Layer-2	**47.6**	**46.9**
Layer-3	47.8	47.0
Layer-4	49.1	48.6
Layer-5	49.0	48.3
Layer-6	86.2	78.2

forcing the features to be domain-invariant from the very first layer interferes with this inherent operation of the network, resulting in lower overall performance. Instead, we should allow the first few layers to contain domain-specific information and only force invariance at a subsequent layer where we expect higher-level knowledge that is independent of the recording conditions.

Moreover, note that when the domain discriminator is based on 6th layer features, learning cannot converge and yields a high error rate. Although this happens only in this case with the chosen hyper-parameters described here (discriminator architecture, learning rate, etc.), we have observed that other choices of hyper-parameters also lead to this phenomenon. Similar difficulties have been reported for GAN training in other applications [84]. In our case, we attribute this to the network's failure in reaching an equilibrium between the domain discriminator and the generator layers. When the generator contains all 6 BLSTM layers (last row of Table 4.3), it significantly outperforms the domain classifier and does not allow it to learn any discriminating information about the domains. This poorly trained discriminator will then return gradients to the recurrent layers which cause learning to diverge completely. In other words, similar to most existing GAN applications, successful training in our case depends on a careful choice of hyper parameters which enables the network to reach an equilibrium between the generator and the domain discriminator. In essence, adversarial

training is an inherently more difficult learning problem, because unlike most other learning problems which seek minimization of a single cost, here we are interested in an equilibrium between two opposing objectives.

4.6 Summary

We presented a novel multi-domain training approach for neural network acoustic models which makes use of environment labels in far-field speech data in order to achieve increased invariance with repsect to the recroding conditions in different rooms. Unlike conventional multi-condition training which combines data from different recording environments into a single set, we consider multi-environment datasets to consist of different domains with slightly different distributions. We presented an analytic study on how a deep network learns to derive environmentally robust features solely based on label sequence supervision. It was shown that the initial layers in a deep network function as a domain separator, mapping data from different rooms into different subspaces. The subsequent layers can then use this encoded domain knowledge to derive robust final representations. This propagation of domain knowledge within the hidden layers was evaluated using a simple domain classifier trained on features from the different hidden layers, which revealed that in practice there is residual domain information even in the last hidden layer, indicating insufficient domain invariance.

Further, it was shown that if the initial layers in an RNN acoustic model are trained adversarially with respect to a domain classifier which recognizes the recording environments, we can enforce better domain invariance and hence a more robust model. The proposed multi-domain adversarial training strategy was evaluated in an end-to-end speech recognition task based on the AMI corpus. It was shown that with a domain classifier based on features from the second hidden layer, a relative improvement of +3.3% can be achieved in character error rate compared to a multi-condition trained baseline which does not use the domain labels

during training. The overall performance improvement with respect to the clean-trained baseline is +25.4%.

Note that although the data domains in our study correspond to actual rooms in the train data, this does not scale to larger datasets which may contain hundreds of different rooms. For a larger number of rooms, the differences in terms of T_{60}, DRR, etc. may be very small, making it impractical to reliably train a domain discriminator to recognize all individual rooms. In such scenarios, the domain discriminator should be trained to recognize groups or classes of rooms which have similar acoustic properties. An example of this is when the available rooms are grouped according to their reverberation times into different groups where each group represents a certain interval for possible values of T_{60}. Alternatively, if the speaker-to-microphone distances are also known for the input utterances, we can define the domains based on approximate DRR value.

Algorithm 2: Iterative CTC and adversarial updates

Input: Features: $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_T^{(i)}]$,
Label sequences: $\mathbf{y}^{(i)} = [y_1^{(i)}, \cdots, y_L^{(i)}]$,
Domains: $d^{(i)}$

while *stop criterion not met* **do**

  for $k = 1, \cdots, n\_minibatches$ **do**

    Sample a mini-batch of M examples $(\mathbf{X}^{(i)}, \mathbf{y}^{(i)}, d^{(i)})$.

    $\mathbf{p}^{(i)} = [p_1^{(i)}, \cdots, p_S^{(i)}] = f_W\big(f_G(\mathbf{X}^{(i)})\big)$.

    $J_{CTC} = \frac{1}{M} \sum_{i=1}^{M} \mathrm{CTC}\Big(\mathbf{p}^{(i)}, \mathbf{y}^{(i)}\Big)$.

    $\boldsymbol{\theta}_W \leftarrow \boldsymbol{\theta}_W - \eta \nabla_{\boldsymbol{\theta}_W} J_{CTC}$.

    $\boldsymbol{\theta}_G \leftarrow \boldsymbol{\theta}_G - \eta \nabla_{\boldsymbol{\theta}_G} J_{CTC}$.

  end

  for $r = 1, \cdots, r_D$ **do**

    for $k = 1, \cdots, n\_minibatches$ **do**

      Sample a mini-batch of M examples $(\mathbf{X}^{(i)}, \mathbf{y}^{(i)}, d^{(i)})$.

      $\mathbf{q}^{(i)} = [q_1^{(i)}, \cdots, q_{N_D}^{(i)}] = f_D\big(f_G(\mathbf{X}^{(i)})\big)$.

      $J_D = -\lambda \frac{1}{M} \sum_{i=1}^{M} \log(q_{d^{(i)}})$.

      $\boldsymbol{\theta}_D \leftarrow \boldsymbol{\theta}_D - \eta \nabla_{\boldsymbol{\theta}_D} J_D$.

    end

  end

  for $r = 1, \cdots, r_G$ **do**

    for $k = 1, \cdots, n\_minibatches$ **do**

      Sample a mini-batch of M examples $(\mathbf{X}^{(i)}, \mathbf{y}^{(i)}, d^{(i)})$.

      $J_A = -\lambda \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N_D} \frac{1}{N_D} \log(q_j)$.

      $\boldsymbol{\theta}_G \leftarrow \boldsymbol{\theta}_G - \eta \nabla_{\boldsymbol{\theta}_G} J_A$.

    end

  end

end

CHAPTER 5

ADAPTATION OF DNN/RNN ACOUSTIC MODELS TO SPECIFIC ENVIRONMENTS

5.1 The adaptation problem[1]

The robustness of speech recognition systems is influenced by the mismatch between the training data and the test conditions encountered in practice. Many existing strategies in robust ASR aim to minimize this mismatch, either by feature processing to make the input features more similar between the two domains, or by training an acoustic model that generalizes better to unseen conditions, e.g. by using a very large and diverse training dataset. However, in practice there is always residual mismatch between a general acoustic model and test data from a specific target environment. The goal in model adaptation is to refine the parameters of a previously trained model to improve performance on a particular test condition. For far-field ASR, model adaptation is useful in two different contexts:

- **Adaptation of clean-trained acoustic models to far-field data:** An acoustic model that is trained on clean close-talking speech suffers from a severe mismatch with reverberant far-field speech. If a small dataset of far-field speech is available, performance can be considerably improved by adapting the model parameters to the far-field features.

- **Adaptation of multi-condition far-field acoustic models to specific target environments:** An acoustic model that is already trained on far-field data exhibits a smaller mismatch with far-field test conditions. However, if a few adaptation examples are available from a specific target environment with particular acoustic properties,

[1] © 2015 ISCA. Reprinted with permission, from S. Mirsamadi and J. H. L. Hansen, A study on deep neural network acoustic model adaptation for robust far-field speech recognition, in Proc. Interspeech, Dresden, Germany, Sep. 6-10, 2015.

model adaptation can be used to further improve performance. This is because the parameters of the original acoustic model are tuned to perform best *on average* accross all recording conditions present in the training data. If the ASR system is to be deployed in a specific room with particular acoustic characteristics, the mismatch can be further reduced by adapting the paramters towards data coming from that particular room.

Model adaptation in general is a well studied problem in ASR [90–92]. However, a considerable majority of existing approaches focus on speaker adaptation which is a fundamentally different problem from environment adaptation. In speaker adaptation we are concerned with speech production differences, while in far-field adaptation we are interested in influences from the recording condition. Morever, the few existing environment adaptation approaches such as Vector Taylor Series (VTS) adaptation [93, 94] and REMOS adaptation [36, 95] are exclusively developed for GMM-HMM models and are not applicable to DNNs. The goal of this chapter is to develop and evaluate different DNN adaptation strategies for the task of environment adaptation in ASR. We will present DNN adaptation strategies which prevent overfitting, and evaluate them for both scenarios described above and for DNN-HMM hybrids as well as end-to-end RNN models.

5.2 Supervision in model adaptation

The transcripts for adaptation data can either be known in advance (supervised adaptation) or obtained by decoding the data using the unadapted model (unsupervised adaptation). In the unsupervised mode, adaptation performance is limited by the accuracy of the obtained labels. In spite of this limitation, in tasks such as speaker adaptation, the number of correct labels in the initial decoding is often adequate to provide reasonable adaptation performance. However, in the case of environmental mismatch such as noise and reverberation, the initial

decoding with the unadapted model has a high error rate and is thus unable to provide a reasonable number of correct labels for adaptation. Furthermore, discriminative estimation of adaptation parameters is known to be more sensitive to the accuracy of the labels compared to the maximum likelihood estimations used for GMMs. As a result, unsupervised DNN adaptation is not able to provide noticeable improvements in scenarios with considerable environmental mismatch. All of the experiments in this chapter use supervised adaptation towards a set of transcribed adaptation data.

5.3 DNN adaptation approaches

Although DNN-based acoustic models provide superior modeling capability compared to traditional GMM-based models, they are particularly difficult to adapt to new conditions. This is due to the very large number of parameters in a deep network which cause overfitting when tuned on limited adaptation data. If we simply run a few more passes of parameter optimization using adaptation data, the model will severely overfit to the new data, essentially erasing most of the previously learned information. To prevent the overfitting problem in adaptation, there are three major categories of solutions:

- **Domain-specific parameter sets:** By introducing a small set of extra parameters dedicated to a specific environment and adapting only those (while keeping the rest of the network fixed), we can achieve effective adaptation on limited data without the risk of overfitting. Alternatively, instead of using new parameters, we can choose a subset of existing parameters to adapt from the original model. In either case, we assume the domain transfer from the original training distribution to the new condition is a simple transformation that can be described by a small number of parameters.

- **Conservative training:** Instead of choosing parameter subsets, we can adapt all DNN parameters in a *conservative* or regularized manner to ensure overfitting does

not occur. To achieve this, a regularized optimization objective should be used which ensures the output distribution provided by the DNN does not radically change as a result of adaptation.

In the following sections, we describe different DNN adaptation methods from the above categories.

5.3.1 Domain-specific linear transformations

The simplest and most effective method for DNN adaptation is to apply an affine transformation to either the input features, activations of a hidden layer, or the output activations of the network. In the case of input features, the approach is referred to as Linear Input Network (LIN), and is similar in form to the feature-space MLLR (fMLLR) [96], which is a common adaptation technique for GMM-HMM models. Note that in spite of this structural similarity, LIN is fundamentally different from fMLLR in that the parameters of the affine transformation are tuned discriminatively using frame-level senone labels of the adaptation data.

Assuming $\mathbf{x}_t$ to be a context-dependent feature vector (concatenation of multiple consecutive frames), LIN applies an affine transformation of the form,

$$\tilde{\mathbf{x}}_t = \mathbf{W}\mathbf{x}_t + \mathbf{b}, \tag{5.1}$$

where $\tilde{\mathbf{x}}_t$ represents the adapted feature vector. The DNN outputs are computed based on this transformed vector and compared to the ground truth label from the adaptation labels. The resulting errors are then back-propagated from the output to the LIN layer and used to update its parameters (the parameters of the original network are kept fixed and are not updated based on these errors).

Given the fact that $\mathbf{x}_t$ is a concatenation of features from multiple frames, the LIN transformation matrix is sometimes constrained to be block-diagonal, with the parameters of the

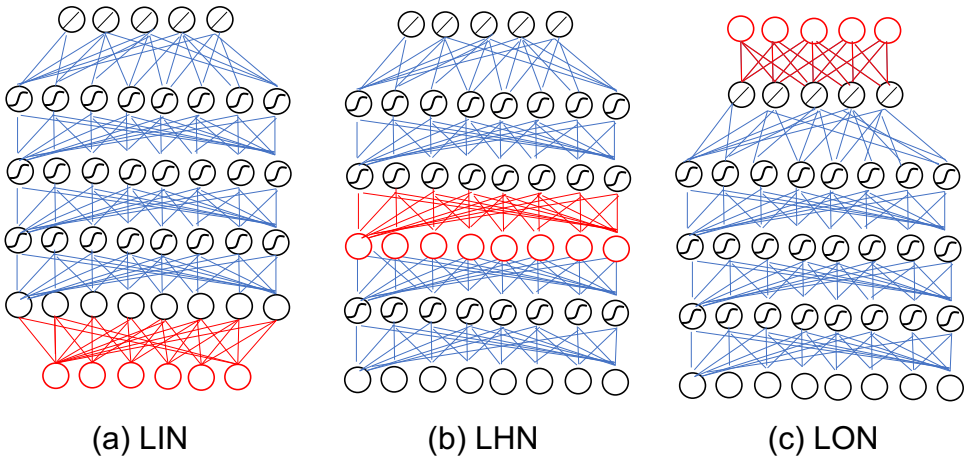


Figure 5.1. Domain-specific linear transformations for environment adaptation. (a) Linear Input Network (LIN). (b) Linear Hidden Network (LHN). (c) Linear Output Network (LON).

diagonal blocks tied together [97]. The resulting adaptation strategy is referred to as feature-discriminative linear regression (fDLR). This frame-specific transformation is reasonable for speaker adaptation where we are interested in updating the features of a single frame individually to match speaker characteristics, and it can lead to improvements specially with small adaptation data because it uses fewer parameters than a fully dense transformation. However, in adaptation for environmental distortions specially in reverberant environments, context information and the correlations between adjacent frames are important information to be used for adaptation. It is therefore desirable in such cases to use the unconstrained (fully dense) transformation matrix $\mathbf{W}$, although this will require more adaptation data for estimating a larger number of parameters.

Similar transformations can be applied to one of the hidden representations or to the final activations of the network. The resulting adaptation strategies are called Linear Hidden Network (LHN) and Linear Output Network (LON), respectively. LHN and LON are equivalent to adding an extra layer with linear activation to the network. Figure 5.1 shows a comparison between these three adaptation methods.

Although LIN, LHN, and LON are similar in nature, their adaptation capability in different tasks may be quite different depending on the amount of available adaptation data

and the nature of the distortion. While the number of adaptation parameters in LHN is determined by the number of nodes in the hidden layers, LIN and LON parameter sizes are dependent on feature dimension and the total number of labels in the output space, respectively. This can influence the amount of adaptation data required for each method. Moreover, the exact position of the adaptation layer is dependent on the nature of the differences between train and test conditions. In particular, the best position for inserting an adaptation layer is where the latent variables (hidden features) are indicative of the particular differences between train and test conditions. This makes the exact position of adaptation layer a task-specific decision. However, for the task of environment adaptation to a specific room, we can use the analysis provided in Section 4.3 on the learning mechanism of DNN acoustic models in order to select the appropriate position of the adaptation layer. We will discuss this choice in more detail in Section 5.6. The experimental observations in Sections 5.4 and 5.5 support the provided guidelines for the choice of adaptation layer position.

5.3.2 Factorized DNN adaptation

Factorized adaptation is an extension of VTS-style adaptation to DNN acoustic models [98]. It is based on the assumption that the final DNN hidden representations for far-field speech can be decomposed into the clean component and linear transformations of noise and channel components. Starting from the input spectral features and assuming only short-time transient channel distortions, the far-field spectral feature vectors can be written as

$$\mathbf{y}_t = \mathbf{x}_t \odot \mathbf{h}_t + \mathbf{n}_t, \tag{5.2}$$

where $\mathbf{y}_t$, $\mathbf{x}_t$, $\mathbf{h}_t$ and $\mathbf{n}_t$ are power spectral features for far-field speech, clean speech, channel transfer function, and additive environment noise, and $\odot$ indicates element-wise multiplication. In the log-spectrum domain, the above relationship can be written as

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{y}}_t - \tilde{\mathbf{h}}_t + \log\left[1 - \exp(\tilde{\mathbf{n}}_t - \tilde{\mathbf{y}}_t)\right], \tag{5.3}$$

where $\tilde{(\cdot)}$ indicates a log-domain variable. This relationship can be expressed concisely as

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{y}}_t + \mathbf{g}(\tilde{\mathbf{y}}_t, \tilde{\mathbf{n}}_t, \tilde{\mathbf{h}}_t), \tag{5.4}$$

where $g(\cdot)$ is the overall nonlinear function that relates $\tilde{\mathbf{y}}_t$ to $\tilde{\mathbf{x}}_t$. In the VTS approach for noise robust ASR [93], a first-order Taylor series approximation of this nonlinear function is used, which effectively expresses the relationship between clean and noisy features as additive distorting factors:

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{y}}_t + \mathbf{A}\tilde{\mathbf{y}}_t + \mathbf{B}\tilde{\mathbf{n}}_t + \mathbf{C}\tilde{\mathbf{h}}_t + \mathbf{d}, \tag{5.5}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are the Jacobian matrices of $\mathbf{g}(\cdot)$ w.r.t. $\mathbf{y}$, $\mathbf{n}$ and $\mathbf{h}$, respectively, and $\mathbf{d}$ is the sum of all constant terms in the VTS expansion. In factorized DNN adaptation, we assume a similar relationship exists between the final hidden features of noisy and clean speech:

$$\hat{\mathbf{v}}_t^L = \mathbf{v}_t^L + \mathbf{A}\mathbf{y}_t + \mathbf{B}\mathbf{n}_t + \mathbf{C}\mathbf{h}_t + \mathbf{d}, \tag{5.6}$$

Here, $\hat{\mathbf{v}}_t^L$ and $\mathbf{v}_t^L$ represent the final hidden representations of the network corresponding to clean and noisy inputs, respectively. The final softmax layer of the network first linearly transforms these features to the output space, followed by a softmax operation to yield posterior probabilities for output symbols:

$$\mathbf{p}_t = \text{softmax}(\mathbf{W}\mathbf{v}_t^L + \mathbf{A}^{'}\mathbf{y}_t + \mathbf{B}^{'}\mathbf{n}_t + \mathbf{C}^{'}\mathbf{h}_t + \mathbf{d}^{'}), \tag{5.7}$$

where $\mathbf{p}_t$ represents the vector of posteriors probabilites for output labels and,

$$\mathbf{A}^{'} = \mathbf{W}\mathbf{A}, \qquad \mathbf{B}^{'} = \mathbf{W}\mathbf{B}, \qquad \mathbf{C}^{'} = \mathbf{W}\mathbf{C}, \qquad \mathbf{d}^{'} = \mathbf{W}\mathbf{d}. \tag{5.8}$$

Equation (5.7) can be rewritten in the compact form

$$\mathbf{p}_t = \text{softmax}(\mathbf{W}^{FA}\mathbf{v}_t^{FA} + \mathbf{d}^{'}), \tag{5.9}$$

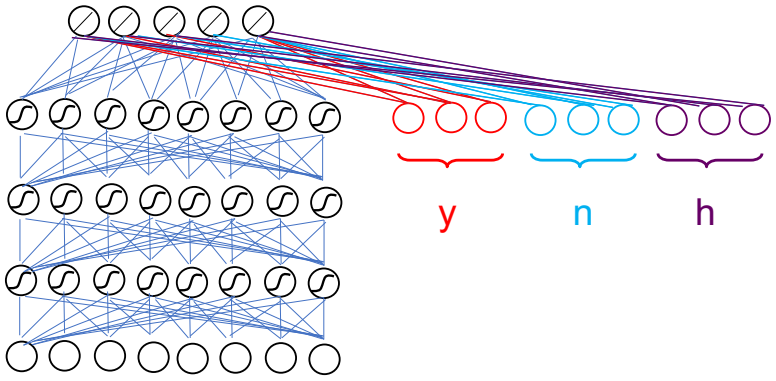


Figure 5.2. Factorized adaptation: Estimated of noise and channel for each frame together with the input noisy feature vector are appended to the final hidden representations.

where,

$$\mathbf{W}^{FA} = [\mathbf{W}, \mathbf{A}', \mathbf{B}', \mathbf{C}'], \tag{5.10}$$

$$\mathbf{v}_t^{FA} = [\mathbf{v}_t^{L^T}, \mathbf{y}_t^T, \mathbf{n}_t^T, \mathbf{h}_t^T]^T. \tag{5.11}$$

According to Equation (5.9), Factorized adaptation is equivalent to concatenating the last hidden layer features with (manual) estimates of noise and channel factors as well as the input noisy feature vector. Figure 5.2 illustrates this equivalence. Assuming we have estimates of the noise and channel components, the adaptation problem consists of estimating the connecting matrices $\mathbf{A}'$, $\mathbf{B}'$, and $\mathbf{C}'$, as well as the offset vector $\mathbf{d}'$. These new parameters are tuned based on adaptation data while keeping the rest of the network fixed.

Note that the provided formulation for factorized adaptation assumes that channel effects are limited to a single frame (hence the use of the model in 5.2). This assumption is valid for microphone or transmission channels, as well as early reverberation. However, as discussed in Section 2.5, late reverberation is a long-term effect which involves multiple time frames. Therefore, factorized adaptation is mostly helpful for additive intereferences (preferrably stationary noise), and has limited performance when applied to the problem of far-field adaptation to reverberant data.

5.3.3 Conservative training

The most straight-forward way of adapting a DNN is to simply adapt all parameters using a few more passes of retraining on the adaptation data. However, given the small size of adaptation data, this would result in overfitting and erase the information learned during training. An effective approach to prevent this overfitting is to force the adapted output distribution to stay close to the unadapted distribution [99]. To achieve this, the Kullback-Leibler (KL) divergence between adapted and unadapted posteriors is added to the optimization criterion:

$$C_{REG} = (1-\rho)C + \rho\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{|S|} p_0\big(s_i(t)|\mathbf{X}\big)\log\frac{p_0\big(s_i(t)|\mathbf{X}\big)}{p\big(s_i(t)|\mathbf{X}\big)}. \tag{5.12}$$

Here, C is the objective for supervised learning (i.e., CTC cost or average cross-entropy error across frames), and $\rho \in [0,1]$ is a trade-off parameter that adjusts the contribution from the regularization term. The terms $p_0\big(s_i(t)|\mathbf{X}\big)$ and $p\big(s_i(t)|\mathbf{X}\big)$ are label distributions for frame t given by the original and adapted models, respectively. The distribution $p_0\big(s_i(t)|\mathbf{X}\big)$ is a constant term (w.r.t. adaptation parameters) that is always provided by the fixed unadapted model. Thus, the terms depending only on $p_0\big(s_i(t)|\mathbf{X}\big)$ can be removed from the objective, yielding the following regularized cost:

$$C_{reg} = (1-\rho)C - \rho\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{|S|} p_0\big(s_i(t)|\mathbf{X}\big)\log p\big(s_i(t)|\mathbf{X}\big) \tag{5.13}$$

Training the network parameters using this regularized cost results in less aggressive updates by penalizing output distributions that are radically different from the original unadapted distribution. The trade-off parameter ρ can be adjusted based on closed-loop performance on a validation set.

For the case where the original supervised learning criterion (C) is frame-level cross-entropy (i.e., in DNN-HMM models where output symbols are senone labels), we can re-write (5.13) in a more compact form by expanding the cross-entropy term (C) which yields:

$$C_{reg} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{s=1}^{|S|} p_{reg}\big(s_i(t)|\mathbf{x}_t\big)\log p\big(s_i(t)|\mathbf{x}_t\big), \tag{5.14}$$

Here, the target distribution $p_{reg}\big(s_i(t)|\mathbf{x}_t\big)$ is a linear combination of the ground-truth label distribution (p_{gt}, given by forced alignment of adaptation data) and the original distribution provided by the unadpted model:

$$p_{reg}\big(s_i(t)|\mathbf{x}_t\big) = (1-\rho)p_{gt}\big(s_i(t)|\mathbf{x}_t\big) + \rho p_0\big(s_i(t)|\mathbf{x}_t\big). \tag{5.15}$$

We are thus effectively replacing the ground-truth hard alignments in the original objective function with a smoothed version which is an interpolation between the original alignments and the distribution given by the unadapted model.

5.4 Experiments on adaptation of clean-trained models to far-field data

5.4.1 System decription and data

We evaluate the discussed DNN adaptation approaches on the single-channel track of Aspire challenge [100]. The data consists of 10-minute audio files of conversational speech from 30 different speakers recorded using far-field microphones in different reverberant and noisy rooms. Half of the utterances from each recording were used as test data and the other half were used for adaptation. A 100-hour subset of the Fisher English corpus [101] was used for training. All experiments use a trigram language model trained on the full Fisher corpus transcripts. The speech features are 13-dimensional Mel-frequency Cepstral coefficients (MFCC) with utterance-based cepstral mean and variance normalization. The input features for the DNN-HMM model are a concatenation of MFCC vectors from a context window of 11 frames. The senone labels for the entire training data are obtained by forced-alignment using an initial GMM-HMM model. For this initial model, dynamic (delta and double-delta) features are used as well, and the concatenated feature vector from 11 context frames was further transformed by Linear Discriminant Analysis (LDA) to a 40-dimensional feature vector. The Kaldi speech recognition toolkit [25] was used for training the GMM-HMM model, with a total number 7716 senones. The DNN model consists of 6 hidden layers of 2048

Table 5.1. Baseline error rates in mismatched conditions (clean-trained models and far-field test data from Aspire challenge)

Acoustic Model	WER(%)
GMM-HMM	74.4
DNN-HMM	62.4

nodes ($\sim$ 37M parameters). The gradient descent optimizations (both for DNN training and adaptation) use a mini-batch size of 256 feature vectors. The DNN parameters are optimized using a learning rate of 0.08 for the first 25 epochs and 0.04 for the rest. Training epochs are stopped when no further improvement is observed on a held-out validation set.

Table 5.1 shows the word error rates of the baseline GMM-HMM and DNN-HMM systems. Note that both models are trained on clean Fisher corpus, and no front-end enhancement has been used for the far-field test data. The resulting error rates are thus high due to the significant underlying mismatch. The DNN-HMM model outperforms the GMM-HMM system by an absolute error difference of 12.0%.

5.4.2 Adaptation results

In this section, we report the recognition results obtained by adapting the clean-trained DNN model to the adaptation utterances selected from Aspire challenge data. For each 10-minute recording, a set of adaptation parameters were estimated based on half of the utterances in the recording (or a subset of them), and used to decode the rest of the utterances. All of the experiments use Stochastic Gradient Descent (SGD) optimization with a minibatch size of 256 and a fixed learning rate of 0.001. In all of the reported results, LHN-i refers to a linear hidden layer added after i'th layer in the network. The transformation matrices in LIN, LHN and LON were initialized as identity matrices, and the biases as zero vectors. For factorized adaptation (FA), compensation matrices $\mathbf{A}'$ and $\mathbf{B}'$ were initialized as zero matrices (we do not use the channel factor). The average feature vector from the first 3 frames of each

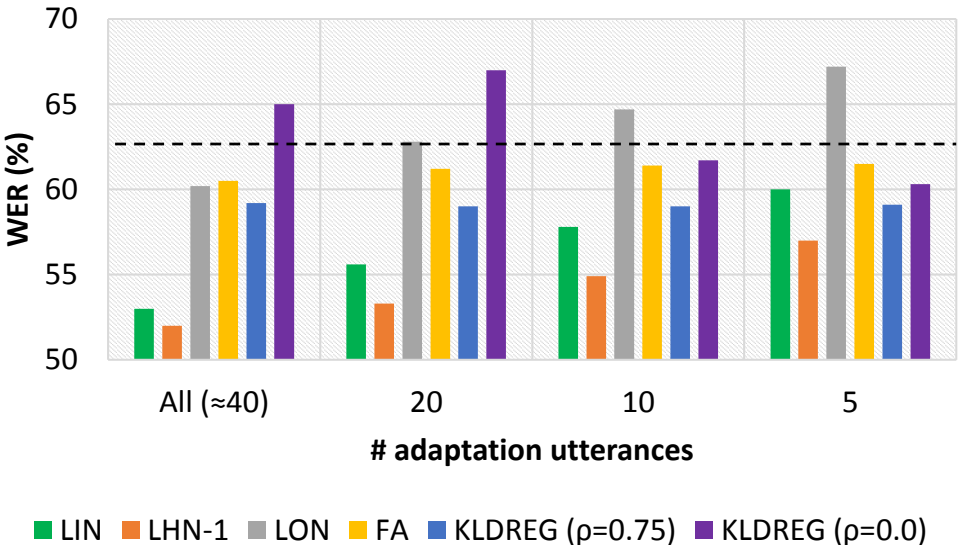


Figure 5.3. Comparison of ASR performance on Aspire data using different adaptation strategies and different amounts of adaptation data. The dashed line indicates performance of the unadapted model.

utterance was used as noise estimate in FA (this assumes a stationary noise across the utterance. Alternatively, more sophisticated strategies such as sparse decomposition [102, 103] can be used to estimate the noise factor). All adaptation experiments are supervised, i.e., the ground-truth transcripts of adaptation utterances have been used to provide labels.

Figure 5.3 compares the word error rates provided by the discussed adaptation strategies using different amounts of adaptation data. The dashed line represents the performance of the unadapted DNN model. It can be observed that the discussed adaptation strategies can provide significant performance gains compared to the clean model. Using the full adaptation dataset, the relative WER improvements range from 16.6% (for LHN-1) to 3.0% (for FA). LIN and LHN have provided the largest overall improvements, particularly when adequate adaptation data is available. LON, on the other hand, has resulted in smaller improvements. This is mainly due to the large output layer size of the DNN (7716 senone targets), which results in a LON transformation matrix with a very large number of parameters which cannot be reliably estimated using the limited adaptation data. Factorized adaptation provides small improvements over the baseline unadapted model, mainly because it is unable to

effectively handle reverberation as discussed in Section 5.3.2. KL-regularized adaptation is able to provide almost consistent improvements even with very low adaptation data sizes, but its overall performance is lower than the linear transormation method in all cases. For comparison, we have also included the results obtained by unregularized adaptation of the DNN parameters ($\rho = 0$), which, as expected, results in poor performance due to overfitting problem. This shows the importance of adding the KLD regularization term (i.e., nonzero value for ρ).

Considering the superior performance of the domain-specific linear transform approach in Figure 5.3, we did a set of experiments to identify the best position in the network for inserting the adaptation layer. The results are depicted in Figure 5.4. It can be observed that for all adaptation data sizes, LHN-1 (i.e. a hidden transform right after the first layer) results in the lowest error rate. In other words, the best domain to perform far-field adaptation is the space of features from the first hidden layer. This is a general observation that we have seen to be consistent across different models and different datasets. We will discuss the reasons for this observation in Section 5.6.

5.5 Experiments on adaptation of multi-condition models to specific target environments

As discussed in Section 5.1, adaptation is useful even with multi-condition models that are already trained on far-field data. Multi-condition training adjusts model parameters to perform best *on average* across all the different rooms and recording conditions within the train data. If recorded adaptation utterances are available from a specific target environment with fixed reverberation properties (T_{60}, DRR, etc.), ASR performance can be further improved by adapting the model towards the test environment. In this section, we provide experimental results to evaluate DNN adaptation performance in such scenarios.

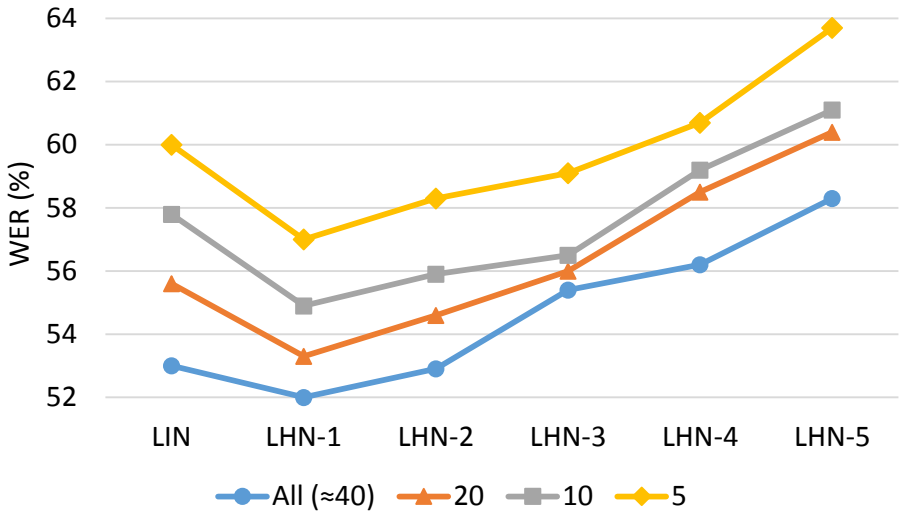


Figure 5.4. Comparison of ASR performance on Aspire data for different positions of the domain-specific layer in the DNN and different amounts of adaptation data from the target domain (5, 10, 20 and 40 utterances).

5.5.1 System decription and data

The experiments in this section are based on the AMI meeting corpus [6] which was described in Section 4.5.1. We remove all utterances containing any overlapped speech frames from both train and test sets, resulting in 30 hours of data for train, and 3.5 hours for each of dev and test sets. We use an end-to-end RNN-CTC model consisting of 3 bi-directional LSTM (BLSTM) layers with 128 cells in each direction, followed by a final softmax layer with 79 outputs representing each of the symbols in our character set plus the blank symbol. We adopt an output space similar to [104], where instead of using a space character, capital letters are used as word delimiters. The input features are 24 dimensional Mel filterbank coefficients extracted from 25 msec frames at a rate of 100 frames per second, and are mean and variance normalized across each speaker. The network parameters are optimized using RMSprop [85] with an inital learning rate of 0.001 and a minibatch size of 20 utterances. We use frame-skipping [87, 105] with a context window of 3 frames to speed up training. Training iterations are stopped when no further improvement is observed on the development

Table 5.2. Character Error Rates on AMI SDM test set. These results use the standard ASR train/dev/test data partitions in [86]

Train Data	Test Data	CER (best-path)	CER (beam-search)
IHM	IHM	37.1	35.8
IHM	SDM	65.8	64.8
SDM	SDM	52.7	51.6

data. Beam search decoding uses a beam width of 10 paths in all cases. All decoding is based on acoustic scores only, using no language or lexicon information.

Table 5.2 shows the obtained character error rates on the AMI far-field (SDM) test set. These results use the standard ASR train/dev/test data partitions in [86], which includes data from all 3 meeeting rooms in the train set, but uses separate meeting sessions (hence different speakers and RIRs) for train and test. The IHM model suffers a sharp performance degradation when presented with far-field SDM test data due to the large mismatch between the acoustic conditions of train and test. Training a model on multi-domain SDM data compensates a significant portion of this degradation, yielding 20% improvement relative to the clean-trained model.

5.5.2 Adaptation results

In this section we use a custom partitioning of AMI data, in which all the data from Edinburgh and TNO meeting rooms (∼25 hours) is used for training, and the IDIAP room data (∼10 hours) is equally split into three subsets for adaptation, development and test. The goal is to investigate adaptation of the original acoustic model trained on Edinburgh and TNO rooms to the acoustic conditions of the IDIAP meeting room. The resuls are provided in Table 5.3. Similar to the previous results on the Aspire challange task discussed in Section 5.4.2 , an intermediate transformation inserted after the first hidden layer is most

effective for adaptation of the deep RNN to the acoustic properties of a new environment. This approach provides 3.5% relative improvement compared to the unadapted model.

Table 5.3. Character Error Rates with different adaption methods

Train Data	Test Data	position of adaptation layer	CER (best-path)	CER (beam-search)
SDM	SDM	None	56.5	55.6
		after input	55.4	54.5
		after layer-1	**54.5**	**53.5**
		after layer-2	55.2	54.2
		after layer-3	55.3	54.4

These results use the custom partitioning of AMI corpus explained in Section 5.5.2 (Train on data from Edinburgh and TNO rooms, and divide the Idiap room data into adaptation, dev, and test sets). Reprinted with permission from [106].

5.6 Determining best position for a domain-specific adaptation layer

We have empirically shown that for far-field adaptation, the best position to insert a domain-specific adaptation layer is after the first hidden layer of the original network. This result holds both for adaptation of clean-trained models to far-field data, as well as adaptation of multi-condition models to specific rooms. Our previous discussion on multi-domain training of deep neural networks in Section 4.3 justifies this observation. The goal in adaptation is to transform the intermediate features such that the resulting distribution resembles the equivalent features from the train data (for which the rest of the network is trained to perform best). In other words, we are interested in a *domain switch* from the new test domain to the domain of training samples. As was pointed out in Section4.3, domain information is maximum in the initial hidden layer of a deep network. Therefore, it is reasonable to perform the domain switch at this stage.

5.7 Summary

We discussed far-field adaptation strategies for DNN-based acoustic models and compared the performance of different DNN adaptation methods. Most existing adaptation studies are either on speaker adaptation or developed for GMM-HMM models. The presented study is intended to address the problem of far-field adaptation for DNNs which are state-of-the-art acoustic models. Two general solutions were considered for the overfitting problem in DNN adaptation. Domain-specific parameter subsets (in the form of linear adaptation layers) can be used to transform intermediate hidden representations to resemble the train data. Alternatively, we can adjust all DNN parameters in a conservative manner by introducing additional regularization terms that ensure sufficient closeness between the original and adapted posteriors. Empirical evaluations both with DNN-HMM hybrids and end-to-end RNNs revealed that the linear transformation approach consisently outperforms other adaptation strategies. Moreover, it was determined that carrying adptation in the space of the first hidden layer features results in best performance. This was explained in light of the observation that adaptation to new room characteristics is more effective in a space where room-dependent features are maximum.

CHAPTER 6

SUMMARY AND CONCLUSIONS

Robustness to far-field distortions is a major challenge in speech recognition due to mismatch caused by the non-stationary and context-dependent nature of these environment distortions. As recent advancements in acoustic modeling have pushed close-talking ASR performance closer to human-level accuracy, far-field ASR has emerged as a natural next milestone in the field which will significantly broaden the application scope. Voice-enabled room/platform systems, distant communication with social/personal robots, and voice interaction in the car, all represent domains where using hand-held microphones (smartphones, etc.) is not possible. This dissertation has focused on the development of solutions both at the front-end and acoustic modeling (back-end) stages to reduce the performance gap between close-talking and far-field ASR. The effectiveness of the proposed solutions have been demonstrated under a variety of both simulated and realistic far-field conditions. In this chapter, we summarize the key thesis contributions and results of this dissertation. We also discuss some open research problems and provide possible future directions for further study.

6.1 Key thesis contributions

Solutions to address the problem of far-field ASR fall under two broad categories of approaches: They either aim to reduce the acoustic mismatch using feature enhancement (single-channel or multi-channel) or model adaptation, or they attempt to create acoustic models that are inherently more robust to existing mismatch. This dissertation has provided solutions from both categories for improved robustness in far-field ASR. Here, we briefly summarize the key contributions made in this dissertation.

Contribution #1: Development of a reverberation-robust multi-channel front-end for distributed microphobe arrays (CNTF algorithm)

Microphone array processing has long been at the center of attention in far-field ASR research. In spite of the variety of existing array processing solutions, most are focused on pre-designed arrays with fixed and known microphone configurations and central processing that is shared among all elements. In other words, they rely on the assumption that the micophones are co-located within a compact array, which makes spatial filtering possible based on phase information between the array elements. Chapter 3 of this dissertation has focused on the alternative case of non-uniform distributed microphones, where the speech signal is captured by independent recording devices in random unknown locations. We introduced a convolutive non-negative tensor factorization algorithm which was able to estimate the clean speech power spectrum by decomposing a tensor of spectrograms from the individual channels into a convolutive combination of channel room impulse responses and a single clean speech estimate. In a clean-trained scenario based on reverberated TIMIT test sentences, four-channel CNTF processing was shown to provide a relative WER improvement of 37.7% over a single-channel case which operates on the microphone with highest DRR. Table 3.2 provides more detailed results on the performance of CNTF algorithm.

The proposed algorithm was shown to provide multiple benefits compared to conventional array processing methods, relaxing many constraints imposed by these methods. By exclusively operating on the spectral amplitudes (discarding phases), the proposed CNTF approach avoids problems resulting from the lack of a shared time reference between the various channels. Essentially, it was shown that the magnitude information alone from spatially distant microphones provides sufficient complementary information to provide a better estimate of the clean speech magnitude spectrum. This is in contrast to conventional array

techniques which primarily rely on phase information [1]. Note that while conventional array processing requires closely-spaced elements, the CNTF approach actually benefits more from spatially distant microphones. For two microphones that are very close to each other, the magnitude spectra are very similar, (there are only phase differences), and thus they provide limited complementary information. Moreover, the proposed algorithm does not require knowledge about the source location, and does not impose any restrictions on the locations of the microphones. It was shown in Section 3.5.3 that CNTF dereverberation is sufficiently robust to unbalanced DRRs among the channels, meaning that if one of the microphones happens to be much farther from the speaker than the others, its contribution to the final estimate of the clean speech will automatically be minimized.

Contribution #2: Analytic study on the learning mechanism of multi-condition trained neural network acoustic models

Multi-condition training of DNN-based acoustic models (specifically RNNs which can effectively model long-term correlations in reverberant speech) has recently gained inreasing popularity to address the problem of far-field ASR. It is now possible to train deep networks based on large amounts speech data collected in various reverberant environments with alternate acoustic properties (T_{60}, DRR, source to microphone distance, etc.). In Chapter 4, by analyzing the propagation of environment-specific (domain) knowledge within the hidden layers, we demonstrated how a multi-condition trained network learns environmental invariance during training. It was shown that the network first extracts environment specific features in the initial front hidden layers, effectively mapping the data from different recording environments into separate subspaces of the hidden representation space. The subsequent layers

[1] Variants of filter-and-sum beamforming can also make use of magnitue information implicitly, but they have no clear separation for the individual contributions of the phase and magnitude components.

then use these encoded domain-related cues to compensate for environment-induced differences, yielding robust output (final layer) representations that are insensitive to recording conditions.

A remarkable observation was the implicit extraction of this additional domain knowledge while the network receives no supervision concerning the environment labels during training. The only supervising information provided to the network is the sequence of phonemes or characters for each utterance. However, the network implicitly learns that in order to predict this label sequence correctly, it should first discover clues about the recording environment.

Contribution #3: Developing an improved multi-domain training approach for DNN acoustic models based on adversarial training with respected to a domain classifier

Conventional multi-condition training discards any available information regarding the recording environments of the training utterances by compiling all data from different environments into a single train set. The expectation is that the network is itself able to derive robust environment-invariant representations in the hidden layers using the supervision based on label sequences. However, it was shown in Section 4.3.2 that in practice, there is residual information concerning the recording rooms in the last hidden layer, indicating that the network has not achieved complete environment invariance. We proposed an improved neural network training strategy in Chapter 4 which makes use of environment labels during training to encourage the derivation of shared hidden representations among the different environments. It was shown that by adjusting the parameters of the initial layers adversarially with respect to a domain recognizer which predicts the environment labels, we can enforce better invariance to the various recording conditions in the subsequent layer representations. The proposed approach was shown to be similar to multi-task learning, where instead of encouraging discriminant features with respect to an auxilliary task, we encourage

invariance to recording environments. The proposed multi-domain approach was shown to provide a relative character error rate reduction of 3.3% and 25.4% using multi-condition trained and clean-trained baseline models, respectively. Table 4.3 provides a full set of results from multi-domain experiments.

Contribution #4: Far-field adaptation of DNN-based acoustic models

Adaptation is very useful in reducing the mismatch between train data and a particular test environment if transcribed utterances are available from the target environment of interest. Most previously existing adaptation studies for ASR have focused on speaker adaptation and have been exclusively developed for GMM-HMM models. In Chapter 5, we have presented adaptation strategies for DNN-HMM hybrid and end-to-end RNN-based acoustic models that are suitable for environment adaptation. These included environment-specific linear hidden transformations, KLD-regularized adaptation, and factorized adaptation. These approaches attempt to use a limited number of transcribed utterances from a specific test environment in order to update a previously trained model to perform better in the target environment without overfitting to the adaptation utterances. The formulated adpatation methods were evaluated in two different scenarios: (i) adaptation of clean-trained models to far-field data, and (ii) Adaptation of multi-condition (far-field trained) models to specific rooms. It was shown that a simple linear hidden transformation inserted into the initial layers, particularly right after the first layer, consistently outperforms other adaptation strategies in both of these scenarios. This approach provided a relative WER reduction of +16.6% on Aspire challenge data (the full set of comparative results are provided in Figures 5.3 and 5.4). The improved gains from transformations inserted in the front layers was justified based on the analysis provided earlier in Section 4.3 which demonstrated how environment invariance is learned by a deep network during training. In particular, since environment-specific information is maximum in the initial hidden layers (most often the first layer), it is easier to carry a

domain-switch at this stage from the test condition to the average acoustic conditions of train data.

6.2 Future work

In this dissertation, we have taken a number of major steps toward reducing the gap between distant-talking and close-talking ASR performance. However, there is still the potential to further improve performance in order to bring far-field error rates closer to the (currently satisfactory) level of close-talking ASR. Here, we list a few possible directions to pursue that are related to the solutions proposed in this dissertation.

Improving robustness of CNTF algorithm to non-stationary additive interefer-ences

The CNTF algorithm discribed in Chapter 3 was mainly developed to improve reverberation robustness using distributed microphone arrays. Nonetheless, the presented formulation based on alpha-beta divergence allows it to be moderately robust to stationary additive noise as well, due to the flexibility provided by alpha and beta parameters for fitting arbitrary distributions. However, the presense of non-stationary additive interference can result in considerable performance degradation in the CNTF algorithm, as the convolutive tensor model in Equation 3.2 will no longer be accurate. Thus, we need a separate mechanism to suppress additive interference on individual channels before employing CNTF to dereverberate and combine channel signals. We studied one possible solution for handling additive noise in distributed arrays in [107], where sparse decomposition of channel spectra was used prior to CNTF processing. Alternatively, DNN-based solutions could be used in a regression setting to remove additive noise if a dataset of parallel noisy/clean utterances is available.

Adversarial multi-domain training using large datasets with hundreds of different recording conditions

The formulation provided in Chapter 4 for adversarial training assumed a dataset of far-field speech recorded in a number of distinct recording rooms. The domain labels in our study were thus the actual room labels attached to each utterance. This original formulation, however, does not scale to larger datasets, where millions of utteranes are available from possibly hundreds of alternate rooms. In such cases, reliable estimation of the domain discriminator parameters is not possible. Moreover, in many cases, the training set is synthetically generated by randomly selecting from a large number of simulated RIRs [108]. In such situations, there are no explicit room labels attached to the utterances. Rather, the training examples are sampled from a continuous distribution of acoustic parameters such as reverberation time (T_{60}), DRR, microphone-to-speaker distance, etc. To use adversarial multi-domain training in such cases, the defined domains should correspond to specific ranges for the acoustic properties of the environment. For example, the training utterances can be roughly categorized to low-DRR, moderate-DRR, and high-DRR conditions. Alternatively, we could divide the range of possible T_{60} values into fixed intervals (e.g., 100 msec intervals spanning the range from 0 to 1 second). The domains in this case would correspond to how reverberant the specific recording environment might be.

Unsupervised DNN adaptation

All of the far-field adaptation approaches discussed in Chapter 5 require accurate human-provided transcripts for adaptation utterances in order to provide sufficient improvement. In practice, it is often difficult to obtain transcribed utterances from the test environment. In contrast, it is fairly easy to collect *unlabaled* data from specific environments after the system is deployed in that environment. Therefore, it is desirable to have unsupervised adaptation

strategies that can provide improvement given only the recorded speech without access to text trancripts.

In the context of speaker adaptation in close-talking ASR, it is possible to first use the initial (unadapted) model to decode the adaptation utterances and obtain labels, and then use these estimated labels as ground-truth during adaptation. However, for far-field adaptation, the accuracy of the labels provided by the initial decoding is not sufficient to yield any overall improvement. Thus, alternative approaches are needed for unsupervised adaptation in far-field scenarios. One possibility is to use unsupervised domain-transfer strategies similar to [78] that are based on adversarial training. The basic idea in such approaches is to employ a particular regularization strategy during training which ensures the distribution of intermediate representations is similar between source and target domain data. If such a common representation is learned between the two domains, reducing the loss on source domain data would also improve target domain performance.

Taken collectively, the contribitions developed in this dissertation have advanced system performance for far-field distance-based ASR, which will have direct impact on speech and language based voice-enabled technologies, in the home, office, and public domains.

REFERENCES

[1] Marco Jeub, Magnus Schafer, and Peter Vary. A binaural room impulse response database for the evaluation of dereverberation algorithms. In *16th International Conference on Digital Signal Processing*, July 2009.

[2] Biing-Hwang Juang and Lawrence R Rabiner. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, 1:67, 2005.

[3] David S. Pallett, Jonathan Fiscus, and John Garofolo. Resource management corpus: September 1992 test set benchmark test results. In *Proceedings of ARPA Microelectronics Technology Office Continuous Speech Recognition Workshop, Stanford, CA*, 1992.

[4] Douglas B. Paul and Janet M. Baker. The design for the wall street journal-based CSR corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, 1992.

[5] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. SWITCHBOARD: telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520 vol.1, Mar. 1992.

[6] Iain McCowan, Jean Carletta, W Kraaij, S Ashby, S Bourban, M Flynn, M Guillemot, T Hain, J Kadlec, V Karaiskos, et al. The AMI meeting corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, volume 88, 2005.

[7] David S. Pallett. A look at NIST's benchmark ASR tests: past, present, and future. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 483–488, Nov. 2003.

[8] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[9] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*, 2016.

[10] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals. Convolutional neural networks for distant speech recognition. *IEEE Signal Processing Letters*, 21(9):1120–1124, 2014.

[11] Geoffrey Hinton, Li Deng, Dong Yu, Goerge E. Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29 (6):82–97, Nov. 2012.

[12] Yajie Miao, Hao Zhang, and Florian Metze. Speaker adaptive training of deep neural network acoustic models using i-vectors. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(11):1938–1949, 2015.

[13] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, and Phil Hall. English conversational telephone speech recognition by humans and machines. *arXiv preprint arXiv:1703.02136*, 2017.

[14] Vijayaditya Peddinti, Vimal Manohar, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur. Far-field ASR without parallel data. In *Proceedings of Interspeech*, 2016.

[15] Yanmin Qian, Tian Tan, and Dong Yu. An investigation into using parallel data for far-field speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5725–5729. IEEE, 2016.

[16] Navid Shokouhi. *Automatic Speaker Recognition and Diarization in Co-channel Speech*. PhD thesis, The University of Texas at Dallas, 2016.

[17] Larry P Heck and Mark Z Mao. Automatic speech recognition of co-channel speech: integrated speaker and speech recognition approach. In *INTERSPEECH*, 2004.

[18] Xianchuan Yu, Dan Hu, and Jindong Xu. *Blind source separation: theory and applications*. John Wiley & Sons, 2013.

[19] Dong Yu and Li Deng. *Automatic speech recognition: A deep learning approach*. Springer, 2014.

[20] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Foreword By-Reddy. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR, 2001.

[21] Matthias Wölfel and John McDonough. *Distant speech recognition*. John Wiley & Sons, 2009.

[22] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[23] Steve J. Young, JJ Odell, and Phil Woodland. Tree based state tying for high accuracy modeling. In *ARPA Workshop on Human Language Technology*, 1994.

[24] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. The HTK book. *Cambridge university engineering department*, 3:175, 2002.

[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011.

[26] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[27] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pages 577–585, 2015.

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

[29] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[30] Awni Y Hannun, Andrew L Maas, Daniel Jurafsky, and Andrew Y Ng. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. *arXiv preprint arXiv:1408.2873*, 2014.

[31] Yajie Miao, Mohammad Gowayyed, and Florian Metze. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 167–174. IEEE, 2015.

[32] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[33] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE, 2016.

[34] Hirokazu Kameoka, Tomohiro Nakatani, and Takuya Yoshioka. Robust speech dereverberation based on non-negativity and sparse nature of speech spectrograms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 45–48, April 2009.

[35] Jan S. Erkelens and Richard Heusdens. Correlation-based and model-based blind single-channel late-reverberation suppression in noisy time-varying acoustical environments. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7):1746–1765, Sept. 2010.

[36] Armin Sehr. *Reverberation Modeling for Robust Distant-Talking Speech Recognition*. PhD thesis, Friedrich-Alexander-Universitat Erlangen-Nurnberg, 2010.

[37] Kenichi Kumatani, John McDonough, and Bhiksha Raj. Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors. *IEEE Signal Processing Magazine*, 29(6):127–140, Nov. 2012.

[38] Michael L Seltzer. *Microphone array processing for robust speech recognition*. PhD thesis, Carnegie Mellon University (CMU), Pittsburgh PA, 2003.

[39] John H. L. Hansen and Xianxian Zhang. Analysis of CFA-BF: Novel combined fixed/adaptive beamforming for robust speech recognition in real car environments. *Speech Communication*, 52(2):134–149, 2010.

[40] Tao Yu and John H. L. Hansen. Automatic beamforming for blind extraction of speech from music environment using variance of spectral flux-inspired criterion. *IEEE Journal of Selected Topics in Signal Processing*, 4(5):785–797, 2010.

[41] Jacob Benesty, Jingdong Chen, and Yiteng Huang. *Microphone array signal processing*, volume 1. Springer Science & Business Media, 2008.

[42] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. *Speech and Audio Processing, IEEE Transactions on*, 2(4):578–589, Oct. 1994.

[43] Seyed Omid Sadjadi and John H.L. Hansen. Hilbert envelope based features for robust speaker identification under reverberant mismatched conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5448–5451, May. 2011.

[44] Herbert Buchner and Walter Kellermann. *Speech Dereverberation*, chapter TRINICON for Dereverberation of Speech and Audio Signals, pages 311–385. Springer-Verlag, 2010.

[45] James R. Hopgood and Peter J.W. Rayner. Blind single channel deconvolution using nonstationary signal processing. *Speech and Audio Processing, IEEE Transactions on*, 11(5):476–488, Sept. 2003.

[46] Tomohiro Nakatani, Takuya Yoshioka, Keisuke Kinoshita, Masato Miyoshi, and Biing-Hwang Juang. Speech dereverberation based on variance-normalized delayed linear prediction. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(7): 1717–1731, Sept. 2010.

[47] Kshitiz Kumar and Richard M. Stern. Environment-invariant compensation for reverberation using linear post-filtering for minimum distortion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4121–4124, March 2008.

[48] Andrew L. Maas, Quoc V. Le, Tyler M. ONeil, Oriol Vinyals, Patrick Nguyen, and Andrew Y. Ng. Environment-invariant compensation for reverberation using linear post-filtering for minimum distortion. In *Interspeech.*, 2012.

[49] Mehrez Souden, Keisuke Kinoshita, Marc Delcroix, and Tomohiro Nakatani. Distributed microphone array processing for speech source separation with classifier fusion. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6, Sept. 2012.

[50] Nobutaka Ono, Hitoshi Kohno, Nobutaka Ito, and Shigeki Sagayama. Blind alignment of asynchronously recorded signals for distributed microphone array. In *Applications of Signal Processing to Audio and Acoustics, WASPAA, IEEE Workshop on*, pages 161–164, Oct. 2009.

[51] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shunichi Amari. *Nonnegative Matrix and Tensor Factorizations : Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, 2009.

[52] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2000.

[53] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Computation*, 21(3):793–830, Mar. 2009.

[54] Raul Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural Computation*, 19(3):780–791, Mar. 2007.

[55] Paris Smaragdis, Cédric Févotte, Gautham J. Mysore, Nasser Mohammadiha, and Matthew Hoffman. Static and dynamic source separation using nonnegative factorizations: A unified view. *Signal Processing Magazine, IEEE*, 31(3):66–75, May 2014.

[56] Andrzej Cichocki, Sergio Cruces, and Shun ichi Amari. Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy*, 13: 134–170, Jan. 2011.

[57] Nasser Mohammadiha, Paris Smaragdis, and Simon Doclo. Joint acoustic and spectral modeling for speech dereverberation using non-negative representations. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4410–4414, April 2015.

[58] Seyedmahdad Mirsamadi and John H.L. Hansen. Multichannel speech dereverberation based on convolutive nonnegative tensor factorization for ASR applications. In *Interspeech*, 2014.

[59] Kshitiz Kumar, Rita Singh, Bhiksha Raj, and Richard Stern. Gammatone sub-band magnitude-domain dereverberation for ASR. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4604–4607, May 2011.

[60] Martin Wolf and Climent Nadeu. Channel selection measures for multi-microphone speech recognition. *Speech Communication*, 57:170–180, Feb. 2014.

[61] Kenichi Kumatani, John McDonough, Jill Fain Lehman, and Bhiksha Raj. Channel selection based on multichannel cross-correlation coefficients for distant speech recognition. In *2011 Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pages 1–6, May 2011.

[62] Aapo Hyv arinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[63] Aapo Hyv arinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51:2499–2512, 2007.

[64] Zhiyun Lu, Zhirong Yang, and Erkki Oja. Selecting beta-divergence for nonnegative matrix factorization by score matching. In *Proceedings of 22nd International Conference on Artificial Neural Networks (ICANN)*, 2012.

[65] Michael L. Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7398–7402, May 2013.

[66] Li Deng, Alex Acero, Mike Plumpe, and Xuedong Huang. Large-vocabulary speech recognition under adverse acoustic environments. In *INTERSPEECH*, pages 806–809, 2000.

[67] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. Highway long short-term memory RNNs for distant speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5755–5759. IEEE, 2016.

[68] Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee. Residual LSTM: Design of a deep recurrent architecture for distant speech recognition. *arXiv preprint arXiv:1701.03360*, 2017.

[69] Yanmin Qian, Tian Tan, and Dong Yu. Neural network based multi-factor aware joint training for robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12):2231–2240, Dec. 2016.

[70] Ritwik Giri, Michael L Seltzer, Jasha Droppo, and Dong Yu. Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5014–5018. IEEE, 2015.

[71] Yajie Miao and Florian Metze. Distance-aware DNNs for robust speech recognition. In *INTERSPEECH*, pages 761–765, 2015.

[72] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. Batch-normalized joint training for DNN-based distant speech recognition. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 28–34, Dec. 2016. doi: 10.1109/SLT.2016.7846241.

[73] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. A network of deep neural networks for distant speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, Mar. 2017.

[74] Richard P. Lippmann, Edward A. Martin, and Douglas B. Paul. Multi-style training for robust isolated-word speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, volume 12, pages 705–708. IEEE, 1987.

[75] John H. L. Hansen. Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. *Speech communication*, 20(1-2):151–173, 1996.

[76] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*, pages 2672–2680, 2014.

[77] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[78] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[79] Yusuke Shinohara. Adversarial multi-task learning of deep neural networks for robust speech recognition. *Interspeech 2016*, pages 2369–2372, 2016.

[80] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[81] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[82] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. Automatic speech emotion recognition using recurrent neural networks with local attention. In *ICASSP*, 2017.

[83] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, pages 1–10, 2015.

[84] Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[85] Geoffrey Hinton. Neural networks for machine learning. Coursera, video lectures., 2012. URL https://www.coursera.org/learn/neural-networks.

[86] The AMI meeting corpus. URL http://groups.inf.ed.ac.uk/ami/corpus/datasets.shtml.

[87] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*, 2015.

[88] Geoffrey Zweig, Chengzhu Yu, Jasha Droppo, and Andreas Stolcke. Advances in all-neural speech recognition. *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017.

[89] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, volume 14, pages 1764–1772, 2014.

[90] Shigeki Sagayama1 Koichi Shinoda, Mitsuru Nakai, and Hiroshi Shimodaira. Analytic methods for acoustic model adaptation: A review. In *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.

[91] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, Apr. 1994.

[92] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2):171 – 185, 1995.

[93] Pedro J. Moreno, Bhiksha Raj, and Richard M. Stern. A vector Taylor series approach for environment-independent speech recognition. In *ICASSP*, volume 2, pages 733–736 vol. 2, May 1996.

[94] Alex Acero, Li Deng, Trausti T Kristjansson, and Jerry Zhang. HMM adaptation using vector taylor series for noisy speech recognition. In *INTERSPEECH*, pages 869–872, 2000.

[95] Armin Sehr, Markus Gardill, and Walter Kellermann. Adapting HMMs of distant-talking ASR systems using feature-domain reverberation models. In *Signal Processing Conference, 2009 17th European*, pages 540–543. IEEE, 2009.

[96] Balakrishnan Varadarajan, Daniel Povey, and Stephen M Chu. Quick FMLLR for speaker adaptation in speech recognition. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4297–4300. IEEE, 2008.

[97] Frank Seide, Gang Li, Xie Chen, and Dong Yu. Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 24–29, Dec. 2011.

[98] Jinyu Li, Jui-Ting Huang, and Yifan Gong. Factorized adaptation for deep neural network. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5537–5541. IEEE, 2014.

[99] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7893–7897. IEEE, 2013.

[100] Mary Harper. The automatic speech recogition in reverberant environments (ASpIRE) challenge. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 547–554. IEEE, 2015.

[101] Christopher Cieri, David Miller, and Kevin Walker. The Fisher corpus: a resource for the next generations of speech-to-text. In *LREC*, volume 4, pages 69–71, 2004.

[102] Jort F. Gemmeke, Tuomas Virtanen, and Antti Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2067–2080, Sept. 2011.

[103] Seyedmahdad Mirsamadi and John H.L. Hansen. Multichannel feature enhancement in distributed microphone arrays for robust distant speech recognition in smart rooms. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 507–512. IEEE, 2014.

[104] Geoffrey Zweig, Chengzhu Yu, Jasha Droppo, and Andreas Stolcke. Advances in all-neural speech recognition. *arXiv preprint arXiv:1609.05935*, 2016.

[105] Vincent Vanhoucke, Matthieu Devin, and Georg Heigold. Multiframe deep neural networks for acoustic modeling. In *ICASSP*, pages 7582–7585. IEEE, 2013.

[106] Seyedmahdad Mirsamadi and John H. L. Hansen. On multi-domain training and adaptation of end-to-end RNN acoustic models for distant speech recognition. In *INTERSPEECH*, 2017.

[107] Seyedmahdad Mirsamadi and John H.L. Hansen. Multichannel feature enhancement in distributed microphone arrays for robust distant speech recognition in smart rooms. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 507–512. IEEE, 2014.

[108] Chanwoo Kim, Ananya Misra, Kean Chin, Thad Hughes, Arun Narayanan, Tara Sainath, and Michiel Bacchiani. Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home. In *Interspeech 2017*. ISCA, 2017.

BIOGRAPHICAL SKETCH

Seyedmahdad Mirsamadi received his BSEE and MSEE degrees in Electrical and Electronic Engineering from Amirkabir University of Technology, Tehran, Iran, in 2009 and 2011, respectively. In 2013, he joined the Center for Robust Speech Systems (CRSS) at The University of Texas at Dallas (UTD), starting his PhD studies under the supervision of Prof. John Hansen. He was a recipient of the best selected paper award in robust speech recognition at IEEE Spoken Language Technology (SLT) Workshop in 2014. His research interests include speech signal processing, speech recognition and enhancement, machine learning and artificial intelligence, data science, and applications of machine learning in speech and audio processing.

CURRICULUM VITAE

Seyedmahdad Mirsamadi

August 2017

Contact Information:

Department of Electrical Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: mirsamadi@utdallas.edu

Educational History:

B.S., Electrical Engineering, Amirkabir University of Technology, 2009
M.S., Electrical Engineering, Amirkabir University of Technology, 2011
Ph.D., Electrical Engineering, The University of Texas at Dallas, 2017

Robust acoustic modeling and front-end design for distant speech recognition
Ph.D. Dissertation
Electrical Engineering Department, The University of Texas at Dallas
Advisor: Dr. John H. L. Hansen

Multiple sound source localization using blind source separation
Master's Thesis
Electrical Engineering Department, Amirkabir University of Technology
Advisors: Dr. Hamid Sheikhzadeh Nadjar, Dr. Amirhossein Rezaie

Employment History:

Research Intern, Microsoft Research (Audio and Acoustics group), Jun-Aug 2015
Research Intern, Microsoft Research (Multimedia, Interaction, and eXperiences (MIX) group), May-Aug 2016

Professional Recognitions and Honors:

Best selected paper award in robust ASR, IEEE Spoken Language Technology (SLT) Workshop, 2014.

Professional Memberships:

Institute of Electrical and Electronics Engineers (IEEE), 2008–present

Journal Papers:

Seyedmahdad Mirsamadi and John H.L. Hansen, "A Generalized Nonnegative Tensor Factorization Approach for Distant Speech Recognition with Distributed Microphones" in IEEE/ACM Transactions on Audio, Speech, and Language Processing, June 2016.

Seyedmahdad Mirsamadi, Shabnam Ghaffarzadegan, Hamid Sheikhzadeh, Seyed Mohammad Ahadi, Amir Hossein Rezaie, "Efficient Frequency Domain Implementation of Noncausal Multichannel Blind Deconvolution for Convolutive Mixtures of Speech" in *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 20, Issue 8, Oct. 2012.

Conference Papers:

Seyedmahdad Mirsamadi and John H.L. Hansen, "On multi-domain training and adaptation of end-to-end RNN acoustic models for distant speech recognition", in *Interspeech 2017*.

Seyedmahdad Mirsamadi, Emad Barsoum and Cha Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention", in *ICASSP 2017*.

Seyedmahdad Mirsamadi and Ivan Tashev, "Causal speech enhancement combining data-driven learning and suppression rule estimation", in *Interspeech 2016*.

Seyedmahdad Mirsamadi and John H.L. Hansen, "A study on deep neural network acoustic model adaptation for robust far-field speech recognition", in *Interspeech 2015*.

Seyedmahdad Mirsamadi and John H.L. Hansen, "Multichannel feature enhancement in distributed microphone arrays for robust distant speech recognition in smart rooms" in *IEEE Spoken Language Technology (SLT) Workshop*, 2014.

Seyedmahdad Mirsamadi and John H.L. Hansen, "Multichannel speech dereverberation based on convolutive nonnegative tensor factorization for ASR applications" in *Interspeech 2014*.