Temporally Varying Weight Regression for Speech Recognition

Shilin Liu

(B. Eng., Zhejiang University)

School of Computing

National University of Singapore

Dissertation submitted to the National University of Singapore

for the degree of Doctor of Philosophy

July 2014

Declaration

This dissertation is the result of my own work conducted at the School of Computing, National University of Singapore. It does not include the outcome of any work done in collaboration, except where stated. It has not been submitted in whole or part for a degree at any other university.

To my best knowledge, the length of this thesis including footnotes and appendices is approximately 40,000 words.

Shilin Liu

_____

Signature

_____

Date

Acknowledgements

First of all, I would like to show my sincere gratitude to my advisor, Dr. SIM Khe Chai, for his countless supervision, discussion and criticism throughout the work of this dissertation. His guidance included from research suggestion, motivation, to scientific writing. He has kept on arranging the weekly meeting up to four years to track my research progress, and discuss challenging problems. 1 hour short weekly meeting has inspired a lot of interesting works into this thesis. He was also providing the right balance of supervision and freedom so that this thesis can be so manifold and fruitful. I would also thank to many anonymous paper reviewers for the constructive comments, which has significantly improved the quality of this thesis. Furthermore, this work could not have been possible without many wonderful open source softwares: HTK toolkit from the Machine Intelligence Laboratory at Cambridge University, Kaldi toolkit created by researchers from Johns Hopkins University, Brno University of Technology and so on, QuickNet from Speech Group in International Computer Science Institute at Berkeley.

I am also very thankful to the National University of Singapore for kindly providing 4 years research scholarship for my degree and many international conference travel grants. I am also very grateful to Dr. SIM Khe Chai for kindly recruiting me as a research assistant under the ARF funded project "Haptic Voice Recognition: Perfecting Voice Input with a Magic Touch". I would also like to thank ISCA, IEEE SPS for providing the conference travel grants.

I also owe my thanks to the members of Computational Linguistic lab led by Prof. NG Hwee Tou. There are too many individuals to acknowledge, but I must thank, in no particular order, WANG Guangsen, LI Bo, WANG Xuancong, WANG Xiaoxuan, WANG Pidong, Lahiru Thilina Samarakoon, LU Wei. They have made the lab an interesting and wonderful place to work in. I also learned a lot of other techniques, careers, experiences from them. In addition, I must also thank my classmates and friends in Singapore, FANG Shunkai, ZHANG Hanwang, FU Qiang, LU Peng, LI Feng, YI Yu, YU Jiangbo, etc. They have organized many interesting and wonderful activities, which enriched my life after working in Singapore.

Finally, I owe my biggest thank to my family in China for their endless support and encouragement over the years. In particular, I would like to thank my girlfriend, LIU Yilian who has always believed in me!

Contents

Summary

Automatic Speech Recognition (ASR) has been one of the most popular research areas in computer science. Many state-of-the-art ASR systems still use the Hidden Markov Model (HMM) for acoustic modelling due to its efficient training and decoding. HMM state output probability of an observation is assumed to be independent of the other states and the surrounding observations. Since temporal correlation between observations exists due to the nature of speech, this assumption is poorly made for speech signal. Although the use of the dynamic parameters and the Gaussian mixture models (GMM) has greatly improved the system performance, implicitly or explicitly modelling the trajectory temporal correlation can potentially improve the ASR systems.

Firstly, an implicit trajectory model called Temporally Varying Weight Regression (TVWR) is proposed in this thesis. Motivated by the success of discriminative training of time-varying mean (fMPE) or variance (pMPE), TVWR aims of modelling the temporal correlation information using the temporally varying GMM weights. In this framework, the time-varying information is represented by the compact phone/state posterior features predicted from the long span acoustic features. The GMM weights are then temporally adjusted through a linear regression of the posterior features. Both maximum likelihood and discriminative training criteria are formulated for parameter estimation.

Secondly, TVWR is investigated for cross-lingual speech recognition. By leveraging on the well-trained foreign recognizers, high quality posteriors can be easily incorporated into TVWR to boost the ASR performance on low-resource languages. In order to take advantages of multiple foreign resources, multi-stream TVWR is also proposed, where multiple sets of posterior features are used to incorporate richer (temporal and spatial) context information. Furthermore, a separate decision tree based state-clustering for the TVWR regression parameters is used to better utilize the more reliable posterior features.

Third, TVWR is investigated as an approach to combine the GMM and the deep neural network (DNN). As reported by various research groups, DNN has been found to consistently outperform GMM and has become the new state-of-the-art for speech recognition. However, many advanced adaptation techniques have been developed for GMM based systems, while it is difficult to devise effective adaptation methods for DNNs. This thesis proposes a novel method of combining the DNN and the GMM using the TVWR framework to take advantage of the superior performance of the DNNs and the robust adaptability of the GMMs. In particular, posterior grouping and sparse regression are proposed to address the issue of incorporating the high dimensional DNN posterior features.

Finally, adaptation and adaptive training of TVWR are investigated for robust speech recognition. In practice, many speech variabilities exist, which will lead to poor recognition performance for mismatched conditions. TVWR has not been formulated to be

robust against those speech variabilities, such as background noises, transmission channels, speakers, etc. The robustness of TVWR can be improved by applying the adaptation and adaptive training techniques, which have been developed for the GMMs. Adaptation aims to change the model parameters to match the test condition using limited supervision data from either the reference or hypothesis. Adaptive training estimates a canonical acoustic model by removing speech variabilities, such that adaptation can be more effective. Both techniques are investigated for the TVWR systems using either the GMM or the DNN-based posterior features. Benchmark tests on the Aurora 4 corpus for robust speech recognition showed that TVWR obtained 21.3% relative improvements over the DNN baseline system and also outperformed the best system in the current literature.

Keywords: Temporally Varying Weight Regression, Trajectory Modelling, Acoustic Modelling, Discriminative Training, Large Vocabulary Continuous Speech Recognition, State Clustering, Sparse Regression, Adaptation, Adaptive Training

List of Acronyms

ADC	Analog-to-digital
AM	Acoustic Model
ASR	Automatic Speech Recognition
BM	Baum Welch
BMM	Buried Markov model
CD	Context Dependent
CI	Context Independent
cFDLR	constrained Feature Discriminant Linear Regression
CMLLR	Constrained Maximum Likelihood Linear Regression
CMN	Cepstral Mean Normalization
CVN	Cepstral Variance Normalization
CMVN	Cepstral Mean&Variance Normalization
CNC	Confusion Network Combination
CNN	Convolutional Neural Network
DBN	Deep Belief Network
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
DPMC	Data-driven PMC
EM	Expectation Maximization
FAHMM	Factor Analyzed HMM
FFT	Fast Fourier Transform
FMLLR	Feature Maximum Likelihood Linear Regression
GMM	Gaussian Mixture Model
GRBM	Gaussian-Bernoulli RBM
HLDA	Heteroscedastic Linear Discriminant Analysis

HMM	Hidden Markov Model
HTK	HMM Toolkit
HTM	Hidden Trajectory Model
KL	Kullback-Leibler divergence
LDA	Linear Discriminant Analysis
LM	Language Model
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum a Posterior
MFCC	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLLT	Maximum Likelihood Linear Transform
MLE	Maximum Likelihood estimation
MLP	Multiple Layer Perceptron
MMI	Maximum Mutual Information
MMSE	Minimum Mean Square Error
MPE	Minimum Phone Error
MLLR	Maximum Likelihood Linear Regression
NAT	Noise Adaptive Training
NN	Neural Network
OOV	Out-of-vocabulary
PER	Phone Error Rate
PCA	Principle Component Analysis
PLP	Perceptual Linear Prediction Coefficients
PMC	Parallel Model Combination
POS	Part-of-speech
SVM	Support Vector Machine
RBM	Restricted Boltzmann Machine
RDLT	Region Dependent Linear Transform
RNN	Recurrent Neural Network
SAT	Speaker Adaptive Training
SD	Speaker Dependent

SI	Speaker Independent
SER	Sentence Error Rate
SLDS	switching linear dynamical System
SNR	Signal-to-noise Ratio
SSM	Sstochastic Segment model
STC	Semi-tied Covariance
STFT	Short Time Fourier Transform
TPMC	Trajectory-based PMC
TVWR	Temporally Varying Weight Regression
VAD	Voice Activity Detector
VTLN	Vocal Tract Length Normalization
VTS	Vector Taylor Series
WER	Word Error Rate
WSJ	Wall Street Journal

List of Publications

1. **Shilin Liu**, Khe Chai Sim. "Joint Adaptation and Adaptive Training of TVWR for Robust Automatic Speech Recognition," accepted by *Interspeech* 2014

2. **Shilin Liu**, Khe Chai Sim. "On Combining DNN and GMM with Unsupervised Speaker Adaptation for Robust Automatic Speech Recognition," published in *ICASSP* 2014

3. **Shilin Liu**, Khe Chai Sim. "Temporally Varying Weight Regression: a Semi-parametric Trajectory Model for Automatic Speech Recognition," published in *IEEE/ACM Transactions on Audio, Speech and Language Processing* 2014

4. **Shilin Liu**, Khe Chai Sim. "Multi-stream Temporally Varying Weight Regression for Cross-lingual Speech Recognition," published in *ASRU* 2013

5. **Shilin Liu**, Khe Chai Sim. "An Investigation of Temporally Varying Weight Regression for Noise Robust Speech Recognition," published in *Interspeech* 2013

6. **Shilin Liu**, Khe Chai Sim. "Parameter Clustering for Temporally Varying Weight Regression for Automatic Speech Recognition," published in *Interspeech* 2013

7. **Shilin Liu**, Khe Chai Sim. "Implicit Trajectory Modelling Using Temporally Varying Weight Regression for Automatic Speech Recognition," published in *ICASSP* 2012

8. Guangsen Wang, Bo Li, **Shilin Liu**, Xuancong Wang, Xiaoxuan Wang and Khe Chai Sim. "Improving Mandarin Predictive Text Input By Augmenting Pinyin Initials with Speech and Tonal Information," published in *ICMI* 2012

9. Khe Chai Sim, **Shilin Liu**. "Semi-parametric Trajectory Modelling Using Temporally Varying Feature Mapping for Speech Recognition," published in *Interspeech* 2010

List of Tables

List of Figures

Chapter 1

Introduction to Speech Recognition

Speech is one of the most convenient communication approaches between humans and machines. When the speech can be correctly recognized by the machine, it can offer many conveniences for our daily life by avoiding tedious typing, for example, IBMs ViaVoice, a desktop dictation system. After applying various natural language processing techniques to analyze the semantic meaning of the recognized speech, many more useful applications can be developed, such as speech translation, and automated call centers. In particular, virtual personal assistant and its variants, such as iPhones Siri [1], Google Now [2], Bing Search have become very popular recently in the mobile phones. These applications can answer questions or execute commands by simply listening to the people.

The first technology behind these interesting applications is Automatic Speech Recognition (ASR) system, which automatically converts a speech waveform to the word sequence or text. Although speech recognition has been studied since 1960s, it has not been solved yet due to many practical challenges, such as speaker, environment, microphone variabilities and so on. On the other hand, as speech varies in length, advanced classic classifiers, such as Support Vector Machine (SVM) [1] and Neural Network (NN) [1] cannot be directly applied for speech recognition. Hence, Hidden Markov Model (HMM) [2] has become the most popular statistic acoustic model for the state-of-the-art ASR systems. Probability density function of the HMM state can be represented by a multivariate Gaussian mixture model (GMM) [3]. A typical state-of-the-art context-dependent GMM-HMM Large Vocabulary Continuous Speech Recognition (LVCSR) [4] system contains tens of thousands of Gaussian components. Therefore, hundreds or thousands hours of training data are needed for the robust estimation. Moreover, high system complexity also increases the computing cost for both training and decoding. In practice, computer clusters and cloud computing may be collaborated for providing recognition service for mobile applications. In this chapter, a brief introduction of some essential components in the ASR system will be presented.

[1] http://www.apple.com/ios/siri/

[2] http://www.google.com/landing/now/

1.1 Statistical Speech Recognition

In this section, speech recognition based on statistical method will be briefly introduced from the system overview to the mathematical problem definition.

1.1.1 System Overview

Figure.1.1 shows a typical example of the ASR system, which consists of several important components. ASR system takes a raw waveform file as input, and produces a most likely transcription or text hidden in this file. The raw waveform file has to be passed into the feature extraction component first. The purpose is to remove as much nuisance information as possible and keep manipulable and discriminable parameterization. Hence, feature extraction is a process of leveraging the feature dimension and resolution. For decades, researchers have engineered many acoustic features, such as Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction Coefficients (PLP). For example, MFCC includes the short time-frequency analysis, filter bank analysis and discrete cosine transform [5]. These coefficients are also referred to as the static parameters, while their derivatives are usually calculated as the dynamic parameters. Concatenation of these static and dynamic parameters becomes the final acoustic feature. Many other advanced techniques also exist for post-processing of these fundamental acoustic features, such as Linear Discriminant Analysis (LDA) [6], Heteroscedastic LDA (HLDA) [7], Multiple Layer Perceptron (MLP) [8] and so on. More details about the feature extraction will be given in the next chapter.

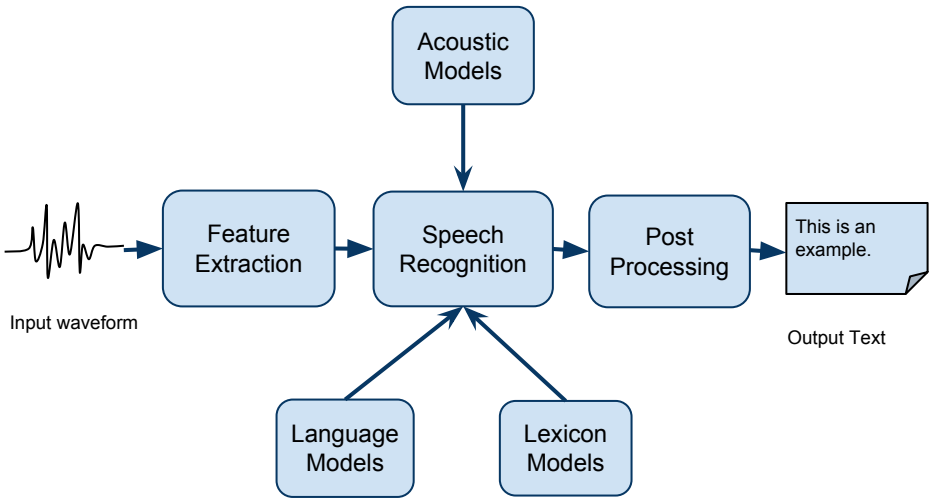

Figure 1.1: Architecture of a typical speech recognition system.

The speech recognition component includes three essential sub-components:

Acoustic Modelling

Acoustic model aims to discriminate different sound unit (such as phoneme, syllable or word) given the observation. Statistic acoustic model is usually employed to learning their characteristics due to existing many speech variabilities. In addition, large amount of speech data with the correct transcription are also needed for supervised training.

Language Modelling

Statistical language model is usually used to calculate the prior probability of a word sequence. It has been widely used in many other areas, such as information retrieval, part-of-speech tagging, etc. In speech recognition, it is primarily used to build the searching network weighted by word transition probability. As the language model complexity grows exponentially with respect to its dependency order, lower order language model is usually applied for full decoding while higher order language model is used for re-scoring.

Lexical Modelling

Lexical model is the connection between acoustic and language models. It is particularly important when the acoustic model is based on the phoneme level, which is the usual case. Lexical model builds the mapping between word and its pronunciation: a phone sequence. If a word has multiple pronunciations, pronunciation probabilities may be modelled for a better recognition. During recognition, vocabulary size is always limited, which can lead to failure of recognition for those out-of-vocabulary (OOV) words.

The post processing component is usually referred to as the system evaluation. In this thesis, I will pay more specific attention on the recognition accuracy, which can be measured by the difference between the recognized hypothesis and the reference. Depending on the purpose of evaluation, different error/distance metrics can be applied: Sentence Error Rate (SER), Word Error Rate (WER), Phone Error Rate (PER). As an utterance can be represented as a sequence of tokens (words or phones), Levenshtein distance has been widely used to calculate WER and PER.

1.1.2 Problem Formulation

Due to the nature of speech recognition, it can be viewed as finding the hidden word sequence of an incoming speech utterance. Mathematically, this problem can be formulated as searching a most likely word sequence given the speech utterance:

$$\hat{\mathcal{W}}_1^N = \arg\max_{\mathcal{W}_1^N} P(\mathcal{W}_1^N | \mathcal{O}_1^T, \boldsymbol{\theta}) \tag{1.1}$$

where $\mathcal{W}_1^N$ is a N-words sequence, $\mathcal{O}_1^T$ is a T-frames observation sequence representing the given utterance, $\boldsymbol{\theta}$ are the underlying model parameters. One biggest challenge here is that N is unknown during recognition. Assuming that the vocabulary size is V, the search space would be V^N. In other words, the ASR system may be infeasible if the recognition algorithm is not carefully designed. Two categories of approaches may be applied to solve this problem [1]:

Probabilistic Generative Model

This approach aims to model the class-conditional densities $P(\mathcal{O}_1^T|\mathcal{W}_1^N)$, as well as the class priors $P(\mathcal{W}_1^N)$, which can be then used to compute the posterior probabilities $p(\mathcal{W}_1^N|\mathcal{O}_1^T)$ through the Bayes' theorem. A typical example is Hidden Markov Model (HMM) [9].

Probabilistic Discriminative Model

This approach directly computes the posterior probability of the class $\mathcal{W}_1^N$ without modelling class-conditional densities. One example for speech recognition is Conditional Random Field [10].

In the case of using the generative model, according to the Bayes' theorem, the conditional probability can be rewritten as:

$$\begin{aligned}\hat{\mathcal{W}}_1^N &= \arg\max_{\mathcal{W}_1^N} \frac{P(\mathcal{O}_1^T|\mathcal{W}_1^N, \boldsymbol{\theta}^{AM})P(\mathcal{W}_1^N|\boldsymbol{\theta}^{LM})}{P(\mathcal{O}_1^T|\boldsymbol{\theta})} \\ &\propto \arg\max_{\mathcal{W}_1^N} P(\mathcal{O}_1^T|\mathcal{W}_1^N, \boldsymbol{\theta}^{AM})P(\mathcal{W}_1^N|\boldsymbol{\theta}^{LM}) \end{aligned} \tag{1.2}$$

where $\boldsymbol{\theta}^{AM}$ and $\boldsymbol{\theta}^{LM}$ are the acoustic model and language model parameters, respectively. Since both N and the alignment between the observation and word sequence are unknown, many famous probabilistic classifiers, such as SVM, NN cannot be applied directly. The ability of modelling varying length of the speech makes the Hidden Markov Model (HMM) as the most popular acoustic model. $P(\mathcal{O}_1^T|\mathcal{W}_1^N, \boldsymbol{\theta}^{AM})$ is also called acoustic model score, which depends on the underlying acoustic model. For instance, if the HMM is applied for acoustic modelling, it will contain the state emission and transition probabilities.

Regarding the language model score, $P(\mathcal{W}_1^N|\boldsymbol{\theta}^{LM})$, further factorization can be performed such as:

$$P(\mathcal{W}_1^N|\boldsymbol{\theta}^{LM}) = P(w_1|\boldsymbol{\theta}^{LM}) \prod_{i=2}^{N} P(w_i|\mathcal{W}_1^{i-1}, \boldsymbol{\theta}^{LM}) \tag{1.3}$$

where w_i is the i-th word of the word sequence, while $\mathcal{W}_1^{i-1}$ is a word sequence occurring before word w_i. In practice, it is difficult to compute $P(w_i|\mathcal{W}_1^{i-1}, \boldsymbol{\theta}^{LM})$ for each i, which requires a lot of training examples and memories. Therefore, approximation is made to obtain a more tractable language model such that

$$P(w_i|\mathcal{W}_1^{i-1}, \boldsymbol{\theta}^{LM}) \approx P(w_i|w_{i-1}, w_{i-2}, ...w_{i-n+1}, \boldsymbol{\theta}^{LM}) \tag{1.4}$$

where n defines the order of dependence on its preceding words, a.k.a. n-gram language model. The typical way to utilize the language model for speech recognition is to use lower order language model to build a smaller search network, generate hypotheses, and then use higher order language model to re-calculate the language model score.

So far, the discussion has assumed that the acoustic and language models are given. Hence, the remaining problem is how to perform training and decoding. Training is to search optimal parameters for $\boldsymbol{\theta}^{AM}, \boldsymbol{\theta}^{LM}$ such that the correct word sequence can have the highest probability given the speech. Supervision based parameter training has to be performed due to the nature of the speech recognition. In addition, training criteria should be carefully chosen by leveraging the training efficiency and recognition accuracy. Decoding is to search the most likely word sequence based on both acoustic and language model scores. As the number of all possible word sequences could be numerically infinite, decoding usually works together with various pruning strategies, such as the beam-search.

In summary, statistical speech recognition includes many essential components, and each of them can have serious impact on the final system performance. To my best knowledge, global optimal solution has not been found for each component yet, therefore there are still many open research topics for each component. In this thesis, the focus will be on acoustic modelling.

1.1.3 Research Problems

Speech recognition research has been going on since the 1960s, but it has not been completely solved yet. This is due to existing many speech related variations during the speech recognition:

- temporal and spatial variations in speech signals (e.g. duration, trajectory)
- inter-speaker variations (e.g. gender, age, non-native speakers)
- intra-speaker variations (e.g. physical body condition)
- channel variations (e.g. microphone, background noise, bandwidths)
- difficulties in modelling syntax and semantics of languages (e.g. words with different part-of-speech (POS) or meanings but with the same pronunciation)
- difficulties in modelling domain information (e.g. literature, finance, science, telephone)
- limited resources for some languages (e.g. limited transcribed training data)

In practice, it is difficult to estimate a speech recognition system to deal with all possible variations. Many applications based on ASR technology work well only on some working conditions. For example, Siri on the iPhone does not work well for non-native English speakers or in a noisy environment. In this thesis, I will focus on dealing with part of above research problems, such as trajectory modelling, speaker variations, channel variations and limited resources issues.

1.2 Thesis Organization

In chapter 2, the most widely used acoustic model, Hidden Markov Models (HMM) will be introduced. First, front-end signal processing for feature extraction is introduced. Next, technical details about formulation, parameter estimation and decoding for GMM-HMM system are discussed. Finally, limitations of HMM are discussed and various advanced techniques are reviewed for solving these limitations, including trajectory modelling, discriminative training, adaptation and adaptive training, deep neural network (DNN) and cross-lingual speech recognition.

In chapter 3, temporally varying weight regression (TVWR) [11, 12] framework is proposed as a new semi-parametric trajectory model for speech recognition. First, a formal probabilistic formulation is given. Next, parameter estimations using both maximum likelihood and discriminative training criteria are introduced. In addition, I-Smoothing is also proposed as an interpolation of two training criteria for a better generalization. Last, experiments are conducted to evaluate the performance based on different training criteria and corpora.

In chapter 4, TVWR [13] is investigated for cross-lingual speech recognition. In particular, temporal and spatial context expansions are proposed to incorporate richer context information for a better recognition accuracy. In addition, a second tree-based state clustering is also proposed for the regression parameters. Experiments are conducted to evaluate this method for cross-lingual speech recognition.

In chapter 5, TVWR is investigated as an approach to combine two state-of-the-arts: GMM and DNN. The goal is to take advantage of the advanced adaptation techniques from GMM and the superior recognition accuracy from DNN. In order to handle the high system complexity of incorporating the high dimensional DNN posteriors, posterior grouping and sparse regression are proposed. Experiments are conducted to evaluate unsupervised speaker adaptation for TVWR using DNN posteriors.

In chapter 6, adaptation and adaptive training are studied for robust TVWR. Adaptation and adaptive training have been widely used to improve the robustness of the speech recognition system. Depending on the types of posteriors features, robust TVWR is investigated via two directions: GMM based posteriors, DNN based posteriors. If GMM based posteriors are used, model compensation can be performed for both the acoustic model and the posterior synthesizer. This approach is also investigated as an approximation of

noise adaptive training. On the other hand, as DNN has been found outperforming GMM for various speech recognition tasks, using DNN posteriors can significantly boost the performance of the TVWR system. Furthermore, joint adaptation and adaptive training of TVWR using DNN based posteriors are investigated.

In chapter 7, the conclusion is drawn and some future works are discussed.

Chapter 2

Acoustic Modelling for Speech Recognition

Hidden Markov Model (HMM) [2] has been widely used as acoustic model for automatic speech recognition for decades. As HMM can subsume the speech data with varying duration, it can be adopted as a generative model to synthesize speech. Due to its probabilistic nature, HMM can also be used as a statistical classifier to perform the speech recognition. After incorporating the Gaussian mixture model (GMM) [3] as the state probability density function, efficient training and decoding algorithms can be derived for GMM/HMM. In this chapter, the attention will be paid on a GMM/HMM recognition system and the advanced state-of-the-art techniques. The important components contain front-end signal processing and parameterization, system evaluation, Viterbi decoding and parameter estimation. Popular state-of-the-art techniques will cover trajectory modelling, discriminative training, adaptation and speaker adaptive training, deep neural networks, cross-lingual speech recognition. Finally, limitations of the current GMM/HMM system and some possible works to circumvent those issues will be discussed.

2.1 Front-end Signal Processing and Feature Extraction

Typically, speech is stored in the waveform file format. Speech recording contains a analog-to-digital conversion (ADC): converting the analog voltage variations caused by air pressure to digital sound. Two key concepts are happening in this process: sampling and quantization, which also serve as the measure of sound quality. When people speak to the microphone, the air pressure is recorded according to a fixed time interval. If a speech waveform is sampled at 16000 times per second, it will have a sampling rate of 16 kHz (kilo Hertz). Higher sampling rate can lead to a better sound quality, but also requires more

storages. Quantization is used to convert the sampled continuous waveform amplitudes to discrete values. Depending on how many bits will be used for the quantization, the accuracy of such approximation will be different. In usual, 8 bits and 16 bits will be used to represent a total of 256 and 65536 possible quantization levels respectively.

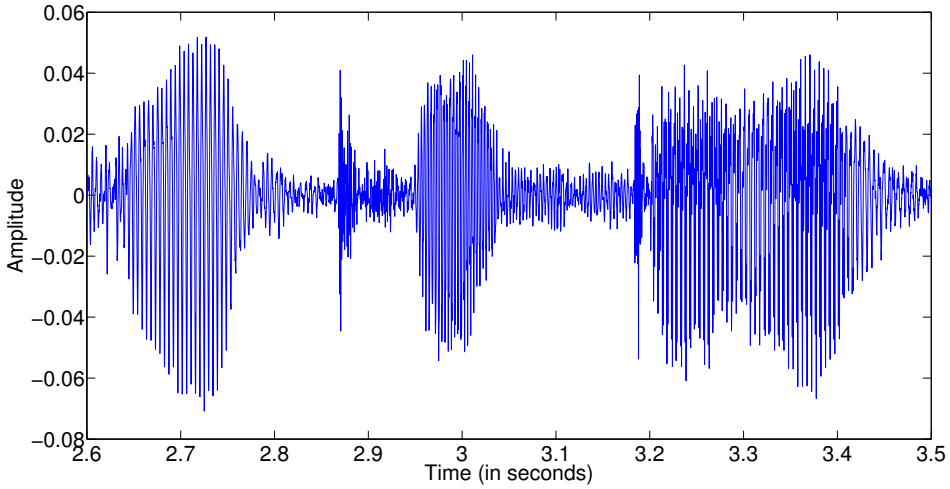

Figure 2.1: An example of waveform with 8 kHz sampling rate.

As the speech waveform contains too much speech-unrelated information, spectral analysis is usually applied, such as Discrete Fourier Transform (DFT) or fast Fourier Transform (FFT). Modern speech parameterization usually employs block processing as shown in Figure. 2.2, which assumes that a short block/frame of samples are quasi-stationary. Frame size is a compromise between the accuracy of time-frequency analysis (needs more samples) and the validness of quasi-stationary assumption (needs fewer samples). Frame shiftting is another factor during block processing, which is used to capture the dynamics of speech. These two factors determine the final number of frames given a speech utterance.

The purpose of block processing is to find a good representation of speech signal, which can be then used to distinguish different speech patterns. As speech pattern is composed of time and frequency, compromise between these two resolutions needs to be made. In order to better understand this concept, spectrogram is introduced. Spectrogram is a two-dimensional visual representation of the Short Time Fourier Transform (STFT) of a time signal. As shown in Figure. 2.3, the spectrogram using 40 ms block size shows better frequency resolution as more samples can be used to calculate more accurate frequencies. However, when compared to the bottom figure using 10 ms block size, the middle one clearly shows worse resolution in the time domain. Except that, there are still many other techniques used during spectral analysis, such as windowing (used for smoothing the edge of block processing), pre-emphasis. Pre-emphasis is used to improve the overall

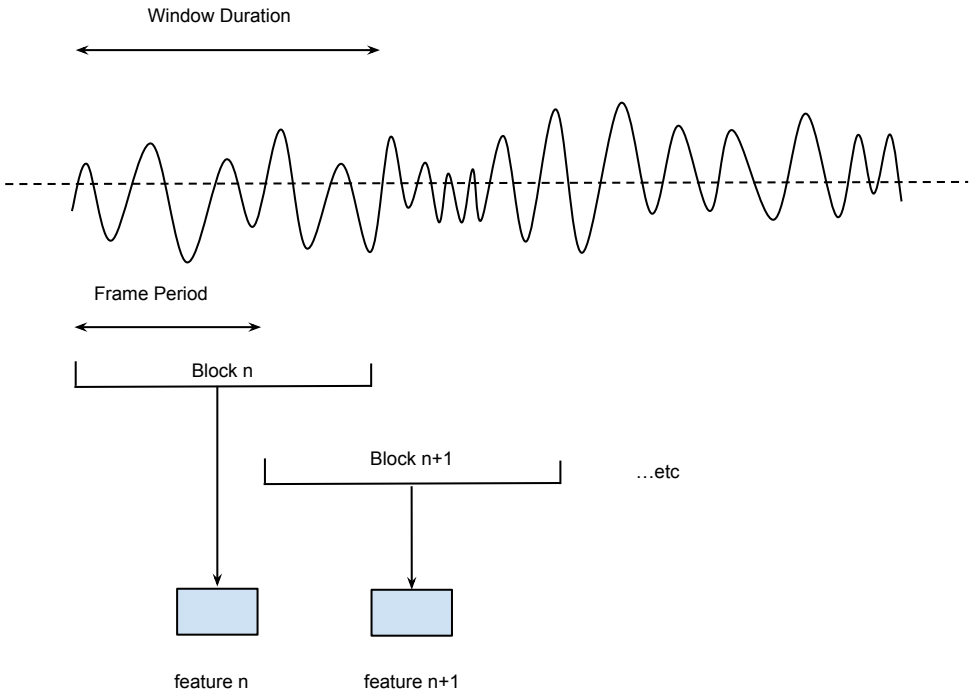

Figure 2.2: An diagram of block processing waveform for feature extraction.

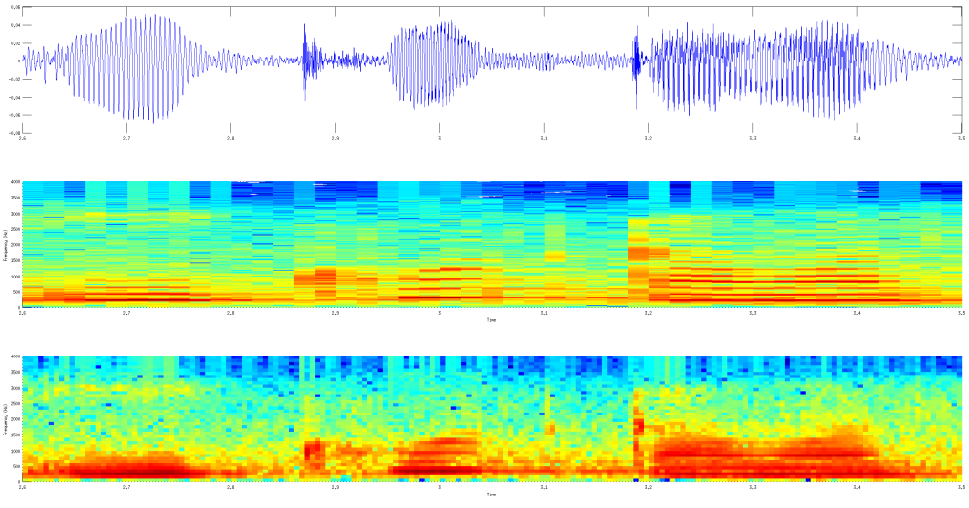

Figure 2.3: Spectrograms using different block size and the same 50% overlapping. Middle: 40 ms block size(better frequency resolution); Bottom: 10 ms block size(better time resolution).

signal-to-noise ratio by adjusting the magnitude of a band of frequencies. In usual, the magnitude of higher frequencies is increased with respect to that of lower frequencies. The formulation of pre-emphasis may be given by $s_n = s_n - \alpha s_{n-1}$, where s_n is the signal at time n, and the pre-emphasis factor α is typically 0.97.

So far, only the raw signal processing and analysis are discussed. Theoretically, the resultant spectral analysis may be used as the acoustic feature for speech processing. However, such acoustic features contain too much redundant information, such as the spectral magnitude by the Short Time Fourier Transform. Alternatively, the spectral magnitude can be represented by filter bank coefficients. In typical, a series of triangular filters are applied and each coefficient corresponding to one filter is the sum of the band passed spectral magnitude. As each frequency is not completely separated, neighbouring filters are defined with overlapping. A filter is usually defined as the percentage of bandpass at a particular location or frequency. In other words, such filter bank coefficient is the weighted sum of band passed spectral magnitude. Now, one most widely used acoustic feature for speech processing (including speech recognition, speaker and language recognition), Mel Frequency Cepstral Coefficients (MFCC) will be introduced. First, the conventional frequency is translated to Mel Frequency by applying a nonlinear mapping below:

$$Mel(f) = 1127 \log(1 + \frac{f}{700.0}) \tag{2.1}$$

Mel scale is the perceptual scale of pitches judged by human listeners to be equal in distance from one another. As shown in Figure. 2.4, a series of Mel filter banks are applied to obtain the Mel filter bank coefficients. In typical, Mel filter bank width increases together with the Mel frequency such that the lower Mel frequency can have a higher resolution. Note that during performing Mel scale translation, the spectral magnitude is still the same as before. In other words, this mapping function is only used for defining the width or distribution of Mel filter banks. Such translation is expected to offer a better discrimination for speech processing, however, it may not be necessary for other purpose. After the Mel filter bank coefficients are ready, the logarithm operation is performed to transform it to log Mel filter bank coefficients, which can be ready as the final features for some applications. However, filter bank coefficients are strongly correlated due to its overlapping, GMM using diagonal covariance matrix has difficulties to model its distribution, while full covariance matrix will significantly increase the system complexity. Alternatively, de-correlation can be performed by truncated Discrete Cosine Transform (DCT), which yields the famous MFCC features. The process to generate MFCC features can be formulated as following equation:

$$c_n = \sqrt{\frac{2.0}{N_{fb}}} \sum_{k=1}^{N_{fb}} \log(m_k) \cos \frac{\pi n(k - 0.5)}{N_{fb}} \quad n = 1, 2, \ldots, N_{mfcc} \tag{2.2}$$

where c_n is the n-th MFCC coefficients, m_k is the k-th Mel filter bank coefficient, N_{fb} and N_{mfcc} are the number of filter bank and MFCC coefficients, respectively. In typical, N_{fb}

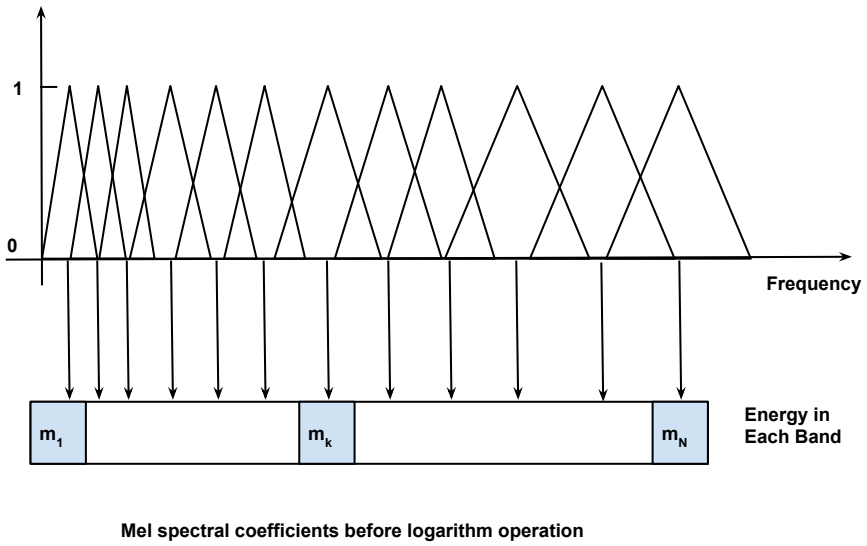

Figure 2.4: Mel filter banks with increasing widths, and Mel spectral coefficients.

is larger than N_{mfcc}, which can be more than twice. In contrast, due to the dimension reduction, this conversion may lose useful information.

Speech is composed of a sequence of correlated acoustic units. In other words, the temporal correlation of successive frames contains rich information to distinguish different acoustic units. One simple way to keep these attributes is to append dynamic parameters to the static parameters produced by Eq-2.2. In some literatures, dynamic parameters are also called differential parameters, as its calculation is one of differential calculation varieties. For example, the first-order differential parameter in speech processing may be given by

$$\Delta c_n = \frac{\sum_{i=1}^{\delta} i(c_{n+i} - c_{n-i})}{2\sum_{i=1}^{\delta} i^2} \tag{2.3}$$

where δ is the delta window, such as 2 in typical. Higher order dynamic parameters can also be obtained by replacing c_n to Δc_n in the right hand side of Eq-2.3. Note that this approach of calculating dynamic parameters can be applied for various parameterization, such as filter bank features, Perceptual Linear Prediction Coefficients (PLP). Typically, up to 3rd or 4th differential parameters followed by subsequent feature projection may be used for speech recognition.

Feature projection is one of effective ways to improve the classification performance, which contains two concepts: feature de-correlation and dimension reduction. It may

be realized by either supervision or un-supervision depending on the availability of data label. The typical example of unsupervised approach is Principle Component Analysis (PCA) [1]. GMM/HMM assumes that the feature element is decorrelated so that diagonal covariance matrix may be applied for efficiency. Feature de-correlation via PCA can make the projected feature more consistent with this assumption. The idea of PCA is to first perform the feature de-correlation by Eigen-decomposition and second pick few directions with largest variations or spreads. In later section, PCA will play an important role of deriving famous tandem system [14].

When the label is known for each data, it is easy to define the objective of feature projection: maximizing the between-class separation and minimizing the within-class spread. One famous example is Fisher's Linear Discriminant, which is a more general formulation of Linear Discriminant Analysis (LDA) [7] without assuming equal covariance matrix between classes. Given a K-classes classification problem, whose k-th distribution is characterized by $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$, the projection vector $\mathbf{w}$ can be obtained by maximizing following Fisher discriminant function:

$$\frac{\delta^2_{between}}{\delta^2_{within}} = \frac{\mathbf{w}^T \boldsymbol{\Sigma}_b \mathbf{w}}{\mathbf{w}^T \boldsymbol{\Sigma}_w \mathbf{w}} \tag{2.4}$$

where $\delta^2_{between}$, δ^2_{within} are the between-class and within-class variances of the projected data respectively,

$$\boldsymbol{\Sigma}_w = \sum_{k=1}^{K} \boldsymbol{\Sigma}_k \tag{2.5}$$

$$\boldsymbol{\Sigma}_b = \frac{1}{K} \sum_{k=1}^{K} (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \tag{2.6}$$

$$\boldsymbol{\mu} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\mu}_k \tag{2.7}$$

It can be shown that the vector $\boldsymbol{w}$ that maximize the above function satisfies the below equation:

$$\boldsymbol{\Sigma}_b \mathbf{w} = \lambda \boldsymbol{\Sigma}_w \mathbf{w} \tag{2.8}$$

or expressed as the standard eigenvalue problem:

$$\boldsymbol{\Sigma}_w^{-1} \boldsymbol{\Sigma}_b \mathbf{w} = \lambda \mathbf{w} \tag{2.9}$$

Therefore, searching the optimum projection vector is translated to find the eigenvectors of $\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\Sigma}_b$. Given K vectors $(\boldsymbol{\mu}_k - \boldsymbol{\mu})$, there are at most $K-1$ independent vectors due to the fact that $\boldsymbol{\mu}$ is a linear combination of $\boldsymbol{\mu}_k$. In other words, the scatter matrix $\boldsymbol{\Sigma}_b$ is at most of rank $K-1$, and there will be a maximum of $K-1$ projection vectors. The final projection vectors can be chosen according to eigenvalues.

Other than above post signal processing techniques, Heteroscedastic Linear Discriminant Analysis (HLDA) [7] is one of the most widely used feature dimension reduction techniques. Semi-tied Covariance (STC) [15] uses a square transformation matrix and hence retains the same dimensionality, while HLDA aims to perform dimension reduction and feature de-correlation, i.e. throwing away dimension which are not useful for classification (also named as the nuisance dimensions). Different from PCA, HLDA is a supervised dimension reduction technique, whose transformation matrix is usually optimized by maximizing the likelihood of training data. In addition to providing better supervised estimation than PCA, HLDA and STC can also estimate a transform for a class of acoustic units instead of a global transformation applied for all.

So far, only simple linear transformation for post signal processing is discussed. There are also many other nonlinear approaches in literatures, such as tandem features [14], discriminatively trained features [16], neural network bottleneck features [17], and so on. More details will be given if related topics are present in later sections. When the acoustic features are ready, we want to discuss how to modulate these features to achieve a better recognition performance.

2.2 Hidden Markov Model (HMM) for Acoustic modelling

Acoustic model is a mathematical representation of an acoustic unit, such as word, syllable or phoneme. As human speech is spontaneous and continuous, boundary information between acoustic units is not present. On the other hand, speaking duration can vary with speakers, contexts, or other conditions, which can lead to different lengths of speech for the same text. In that case, classic classifiers like Support Vector Machine (SVM) and Neural Network (NN) [1] cannot be applied directly, as both require a fixed length of speech. In the next section, Hidden Markov Model will be introduced to solve these problems.

2.2.1 HMM Formulation

The acoustic unit in speech recognition is typically a phone or syllable. As each phone unit can have various duration due to its context, speaker or other environments, it is impossible to build a acoustic model for each phone with different duration. In other words, a candidate acoustic model should have the capability of subsuming varying length of acoustic features. Hidden Markov Model (HMM) is a probabilistic graphical model, whose state can consume varying length of observations such that it becomes a most widely used acoustic model for speech recognition. In a Hidden Markov Model, the state sequence corresponding to the visible observation sequence is unknown (or hidden). HMM

is also known as a finite-state transducer, which can transduce a sequence of observations to a sequence of states. If each phone unit is represented by an HMM, a series of HMMs will be able to transduce the observation sequence to the phone sequence, which later can be translated to the word sequence according to the dictionary. In typical, a phone unit is modelled by a 5-states left-to-right HMM, such as shown in Figure. 2.5:

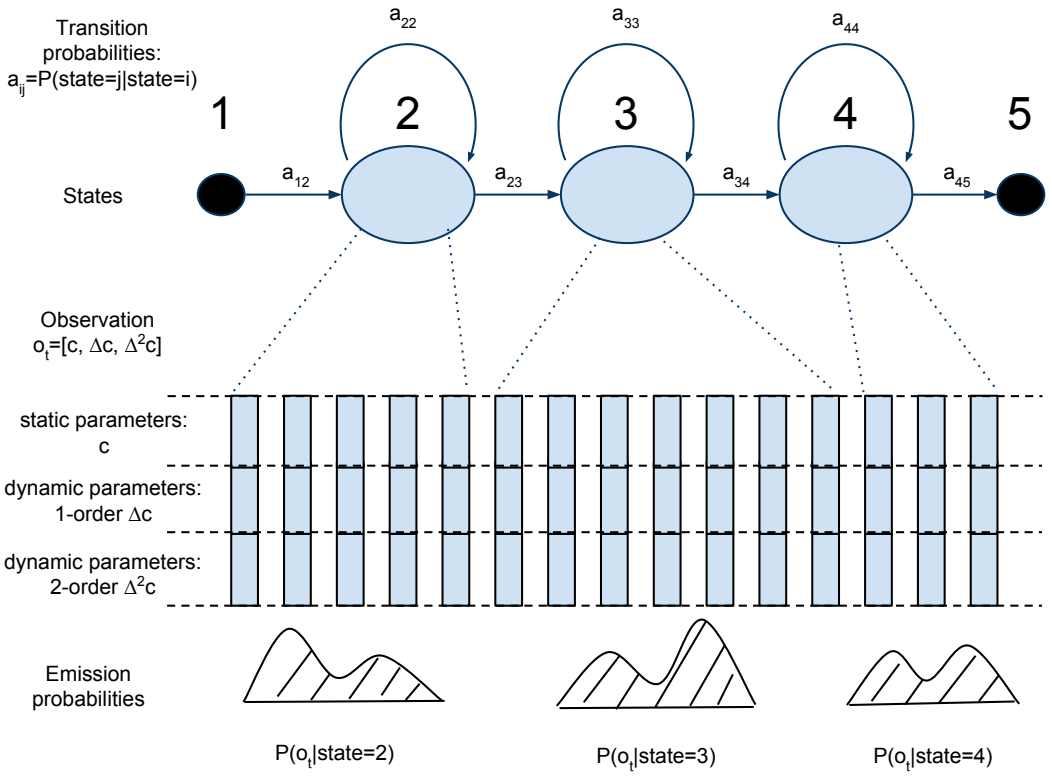

Figure 2.5: A left-to-right model topology of HMM for acoustic modelling

Although HMM can take both discrete feature and continuous feature, the latter one is assumed to be the default in this thesis. As an example, Mel-Frequency Cepstral Coefficients (MFCC) are used as the static parameters, and dynamic parameters can be obtained as the derivative of those static parameters with respect to time [5], and the concatenation of static and dynamic parameters forms the final feature or observation. As a typical graphical model, HMM is composed of two elements:

node A node is used to represent the hidden state in HMM, which has two types for acoustic modelling, emission states (i.e. 2, 3, 4) and non-emission states (i.e. 1, 5). Emission states subsume observations with probabilities, while non-emission states indicate the entry and exit of HMM. Non-emission states are useful for concatenating multiple HMMs as a word. In the case of continuous observations, the state emission probability is actually modelled as probability density function.

arc A arc is used to indicate the possible connection between states. The connection is usually weighted by transition probability in HMM. Typically, the state transition probability is modelled as discrete probability.

These two elements define the topology and complexity of HMM. Higher order HMM [18, 19, 20] may lead to better modelling power, but it requires more training data and may slow down both training and decoding. In a state-of-the-art ASR system, tens of thousands of context dependent phones, i.e. triphone(e.g. a-b+c, where a and c are the left and right context phone of central phone b) or quinphone(e.g. a%b-c+d), are usually employed. In order to make the whole system tractable, two fundamental assumptions are made for HMM used for acoustic modelling:

Instantaneous first-order transition: The probability of making a transition to the next state is independent of other states, given the current state.

Conditional independence assumption: The probability of observing a observation at current time is independent of other observations and states, given the current state.

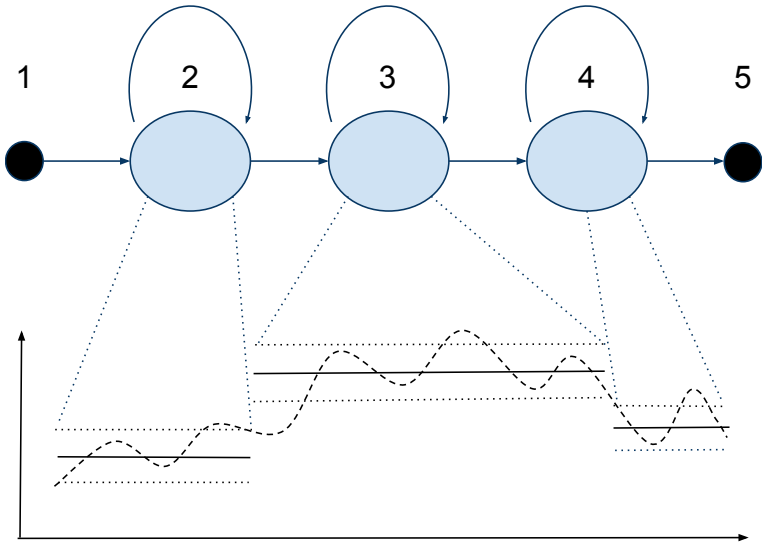

Figure 2.6: A piece-wise stationary process in conventional HMM

According to these two assumptions, a piece-wise stationary process will be retained, as illustrated in Figure. 2.6. In Figure. 2.6, the solid line stands for the mean sequence of one dimensional observation sequence, which is denoted by dashed line. The dotted lines below and above the mean sequence describe the spread of the observation distribution, i.e. the standard deviation. In the following discussion, the mean sequence is also referred

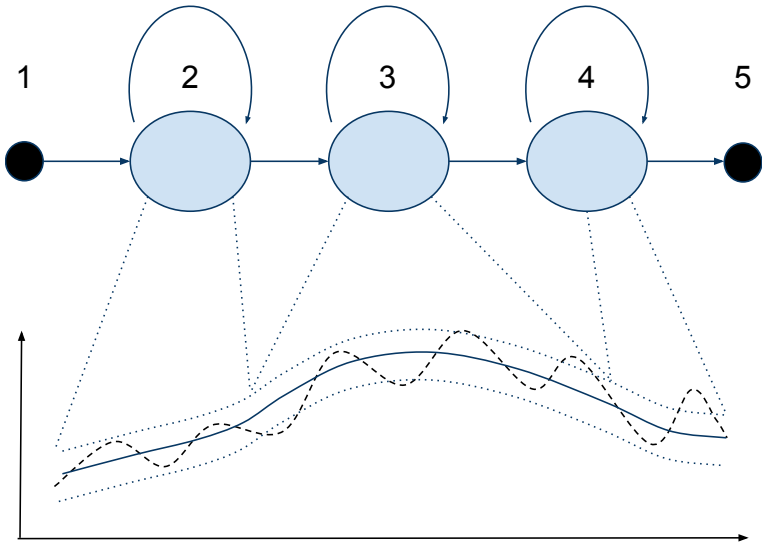

Figure 2.7: A better trajectory representation of speech utterance

to as the trajectory of speech signal. The trajectory shown in Figure . 2.6 is also well known as the piece-wise constant trajectory. This piece-wise constant trajectory is clearly not a good representation of the speech. Instead of using one mean within the state, Gaussian mixture model (GMM) can allow multiple means within the state for a better resolution. This thesis pays more attention on how to solve the limitation caused by the second assumption.

In order to simplify the subsequent discussion, every element of HMM for acoustic modelling is formulated in a more mathematic fashion. First, basic notations for speech processing are introduced:

- $O_1^T = \{\mathbf{o}_t : 1 \leq t \leq T\}$, a sequence of T observations
- $Q_0^{T+1} = \{q_0 = 1, q_{T+1} = S, q_t = j : 1 \leq j \leq S\}$, a sequence of HMM states
- $\mathbf{A} = \{a_{ij} : 1 \leq i, j \leq S\}$, a set of state transition probabilities
- $\mathbf{B} = \{b_j(\mathbf{o}_t) : 1 < j < S, 1 \leq t \leq T\}$, a set of state emission probabilities

where S is the total number HMM states, $\mathbf{o}_t$ is the observation at t, q_t is the state at t, including both the entry, q_0 and exit, q_{T+1} states. Now, the two HMM fundamental assumptions can be expressed as:

$$\text{Instantaneous first-order transition:} P(q_{t+1}|Q_0^{t-1}, q_t) = P(q_{t+1}|q_t) \tag{2.10}$$

$$\text{Conditional independence assumption:} P(o_t|O_1^{t-1}, Q_0^{t-1}, q_t) = P(\mathbf{o}_t|q_t) \tag{2.11}$$

where $O_1^{t-1} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{t-1}\}$ and $Q_0^{t-1} = \{q_0, q_1, \ldots, q_{t-1}\}$. As can be seen, after introducing these two assumptions, the system is significantly simplified. In other words, the total number of parameters for acoustic modelling is greatly reduced.

So far, the fundamental elements of HMM have been introduced. In order to make HMM applicable for speech recognition, the transition matrix $\mathbf{A}$ and emission probabilities $\mathbf{B}$ have to be modelled. Transition matrix simply contains the discrete probabilities, while emission probabilities of multivariate continuous observations may be modelled by Gaussian mixture model. Before applying acoustic model for speech recognition, three below questions have to be resolved:

- Evaluation: How well the model fits to the observations
- Decoding: How to discover the hidden state sequence (or transcription) that generats the observations
- Estimation: How to train the model parameters under certain criteria

In the subsequent sections, these three questions will be discussed in details.

2.2.2 HMM Evaluation: Forward Recursion

This task aims to calculate the probability of observations given the model and transcription, i.e. $P(O_1^T|\mathbf{\Lambda}, \mathcal{W}_1^N)$, where $\mathbf{\Lambda} = \{\mathbf{A}, \mathbf{B}\}$ represents all the model parameters. For convenient notations, the dependency on the transcription, $\mathcal{W}_1^N$ is ignored in the following discussion.

Since the alignment between the state and observation sequence is unknown, the likelihood may be calculated using the marginalization:

$$P(O_1^T|\mathbf{\Lambda}) = \sum_{Q_0^{T+1}} P(O_1^T|Q_1^T, \mathbf{\Lambda}) P(Q_0^{T+1}|\mathbf{\Lambda}) = \prod_{t=1}^{T} \sum_{i,j} P(\mathbf{o}_t|q_t = j) P(q_t = j|q_{t-1} = i) \tag{2.12}$$

However, it is very expensive to directly evaluate this equation due to existing too many possible state sequences. Alternatively, a recursive algorithm has been developed for efficient computing, which is **forward recursion** part of **Forward-Backward algorithm**. Based on the HMM assumptions defined in Eq-2.10 and Eq-2.11, the forward probability

can be simplified as

$$\begin{aligned}\alpha_j(t) &= p(O_1^t, q_t = j|\mathbf{\Lambda}) \qquad (2.13)\\ &= \sum_{i=1}^{S-1} p(\mathbf{o}_t, O_1^{t-1}, q_t = j, q_{t-1} = i|\mathbf{\Lambda})\\ &= p(\mathbf{o}_t|q_t = j, \mathbf{\Lambda}) \sum_{i=1}^{S-1} P(q_t = j|q_{t-1} = i) p(O_1^{t-1}, q_{t-1} = i|\mathbf{\Lambda})\\ &= b_j(\mathbf{o}_t) \sum_{i=1}^{S-1} a_{ij}\alpha_i(t-1) \qquad 1 \le t \le T, 1 < j < S \qquad (2.14)\end{aligned}$$

where

$$\alpha_j(0) = \begin{cases} 1 & \text{for } j = 1 \\ 0 & \text{otherwise} \end{cases} \qquad (2.15)$$

The quantity $\alpha_j(t)$ can be viewed as the partial probability of that the state is j at time t given all possible partial state sequences before t. Thus, the likelihood given the entire observation sequence, O_1^T, can be obtained as

$$p(O_1^T|\mathbf{\Lambda}) = \alpha_S(T+1) = \sum_{i=1}^{S-1} a_{iS}\alpha_i(T) \qquad (2.16)$$

Likelihood is a very important quantity, which can be not only used to express how the model fits the observations, but also a guidance of the training process.

2.2.3 HMM Decoding: Viterbi Algorithm

The objective of speech recognition is to search the hidden word sequence of the input speech utterance. Since the basic acoustic model unit is phone and each phone contains a state sequence, searching the hidden state sequence is the first step of speech recognition. Given an observation sequence, O_1^T, searching the most likely state sequence can be formulated as following function:

$$\hat{\mathbb{Q}}_1^T = \arg\max_{Q_1^T} p(O_1^T, Q_1^T, q_0 = 1, q_{T+1} = S|\mathbf{\Lambda}) \qquad (2.17)$$

A well-known **Viterbi algorithm**, one kind of *dynamic programming* algorithms, can be used to solve this problem efficiently. The main idea is to recursively search the best partial state sequence, which can be formulated as following recursion:

$$\begin{aligned}v_j(t) &= \max_{Q_1^{t-1}} p(O_1^t, Q_1^{t-1}, q_0 = 1, q_t = j|\mathbf{\Lambda})\\ &= \max_{Q_1^{t-2}, 1 \le i < S} p(O_1^t, Q_1^{t-2}, q_0 = 1, q_{t-1} = i, q_t = j|\Lambda)\\ &= p(\mathbf{o}_t|q_t = j) \max_{1 \le i < S} p(q_t = j|q_{t-1} = i) \max_{Q_1^{t-2}} p(O_1^t, Q_1^{t-2}, q_0 = 1, q_{t-1} = i|\mathbf{\Lambda})\\ &= b_j(\mathbf{o}_t) \max_{1 \le i < S} a_{ij} v_i(t-1) \qquad (2.18)\end{aligned}$$

where

$$v_j(0) = \begin{cases} 1 & \text{for } j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

Here $v_j(t)$ denotes the likelihood of the best partial state sequence given the partial observations, O_1^t. However, the decoding process cares more about the best states sequence instead of the likelihood. Therefore, it is important to remember the best previous state for current frame such that the best state sequence can be traced back at the end of utterance. Specifically, a quantity is introduced to achieve this objective:

$$q_j^{max}(t) = \arg \max_{1 \leq i < S} a_{ij} v_i(t-1) \tag{2.20}$$

which denotes the state giving the best partial likelihood at time t. After this process goes through all the observations, trace back is used to restore the best state sequence. The Viterbi algorithm can be viewed as a full search decoding. That means the likelihood for each possible state sequence is calculated, which can cost a lot of computing resources. In practice, a decoding network is usually expanded based on the language model and lexicon model such that those very unlikely partial state sequence can be ignored. At the same time, the decoding result will be the final word sequence instead of intermediate state sequence. On the other hand, beam search by using a beam width during incremental decoding can also help to save the computing cost. This is achieved by ignoring those paths whose likelihood is lower than the defined threshold.

2.2.4 HMM Estimation: Maximum Likelihood

As HMM is a parametric model, parameter estimation is one very important part for acoustic modelling. HMM contains two groups of parameters: transition probabilities and emission probabilities. Typically, transition probabilities are modelled as discrete probabilities, while emission probabilities are modelled by Gaussian mixture models (GMM). In this section, maximum likelihood criterion is introduced to estimate model parameters. In practice, log-likelihood of the model given the observation is maximized as follows:

$$\mathcal{L}(\mathbf{\Lambda}|\mathbf{O}_1^T) = \log p(O_1^T|\mathbf{\Lambda}) = \log \sum_{Q_0^{T+1}} p(O_1^T, Q_0^{T+1}|\mathbf{\Lambda}) \tag{2.21}$$

Note that the dependency on the transcription or word sequence, $\mathcal{W}_1^T$ is ignored for convenient notations. Due to the summation operation within the logarithm function, it becomes difficult to directly optimize such nonlinear function. Alternatively, an efficient algorithm, called **Baum-Welch** algorithm(a.k.a **Forward-backward** algorithm), was developed [21], which is also an example of **Expectation-Maximisation** (EM) algorithms.

Instead of directly maximizing the log-likelihood function, an auxiliary function is introduced such that increasing it can guarantee the increase of the original function. It

is actually a strict lower bound function of likelihood function, and their relationship can be formulated as following inequality:

$$\mathcal{L}(\mathbf{\Lambda}) - \mathcal{L}(\hat{\mathbf{\Lambda}}) \geq \mathcal{Q}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}) - \mathcal{Q}(\hat{\mathbf{\Lambda}}, \hat{\mathbf{\Lambda}}) \tag{2.22}$$

where the auxiliary function is given as

$$\begin{aligned}\mathcal{Q}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}) &= E_{\hat{\mathbf{\Lambda}}}[\log p(O_1^T, Q_0^{T+1}|\mathbf{\Lambda}] \\ &= \sum_{Q_0^{T+1}} p(Q_0^{T+1}|O_1^T, \hat{\mathbf{\Lambda}})\Big(\log p(Q_0^{T+1}|a_{ij}) + \log p(O_1^T|Q_0^{T+1}, \mathbf{\Lambda}^{obs})\Big)\end{aligned} \tag{2.23}$$

$\hat{\mathbf{\Lambda}}$ represents the current model, which is used to estimate the posteriors of state sequence given the observation sequence, and Λ^{obs} contains all the parameters related to the emission probabilities.

The next task is to calculate the posterior probability of state sequence given the observation sequence and model parameters, $p(Q_0^{T+1}|O_1^T, \hat{\mathbf{\Lambda}})$. Based on the Bayes' theorem, the posterior calculation can be split into two parts:

$$p(Q_0^{T+1}|O_1^T, \hat{\mathbf{\Lambda}}) = \frac{p(Q_0^{T+1}, O_1^T|\hat{\mathbf{\Lambda}})}{p(O_1^T|\hat{\mathbf{\Lambda}})} \tag{2.24}$$

where the likelihood of the model $p(O_1^T|\hat{\mathbf{\Lambda}})$ can be calculated according to the forward recursion. The remaining part is the joint probability of the state and observation sequence:

$$\begin{aligned}p(Q_0^{T+1}, O_1^T|\hat{\mathbf{\Lambda}}) &= \sum_{j=1}^{S} p(Q_0^{t-1}, q_t = j, Q_{t+1}^{T+1}, O_1^T|\hat{\mathbf{\Lambda}}) \\ &= \sum_{j=1}^{S} p(Q_0^{t-1}, q_t = j, O_1^t|\hat{\mathbf{\Lambda}}) p(Q_{t+1}^{T+1}, O_{t+1}^T|q_t = j, \hat{\mathbf{\Lambda}}) \\ &= \sum_{j=1}^{S} \alpha_j(t)\beta_j(t) \qquad \forall t\end{aligned} \tag{2.25}$$

where $\alpha_j(t)$ and $\beta_j(t)$ are the **forward** and **backward** probabilities, respectively. Similar to the forward probabilities, $\alpha_j(t)$, introduced previously for HMM evaluation, the backward probabilities, $\beta_j(t)$ can also be calculated recursively:

$$\begin{aligned}\beta_j(t) &= p(Q_{t+1}^{T+1}, O_{t+1}^T|q_t = j, \hat{\mathbf{\Lambda}}) \\ &= \sum_{i=1}^{S-1} p(o_{t+1}|q_{t+1} = i, \Lambda) p(q_{t+1} = i|q_t = j) p(O_{t+2}^T|q_t = j, \Lambda) \\ &= \sum_{i=1}^{S-1} b_i(o_{t+1}) a_{ji} \beta_i(t+1) \qquad 1 \leq t \leq T, 1 < j < S\end{aligned} \tag{2.26}$$

where

$$\beta_j(T) = a_{jS} \tag{2.27}$$

$$\beta_S(T+1) = 1 \tag{2.28}$$

Two optimizing problems can be defined to estimate the parameters of transition and emission probabilities, respectively:

Transition probabilities:

When optimizing transition probabilities, the problem in equation (2.23) becomes maximizing the following function:

$$\mathcal{Q}(a_{ij}, \hat{a}_{ij}) = K_{trans} + \sum_{t=1}^{T}\sum_{i=1}^{S}\sum_{i=1}^{S} \gamma_{ij}(t) \log a_{ij} \qquad s.t. \sum_{j=1}^{S} a_{ij} = 1$$

where K_{trans} is a constant subsuming terms independent of the transition probabilities, and

$$\gamma_{ij}(t) = \frac{p(q_{t-1} = i, q_t = j, O_1^T|\hat{\mathbf{\Lambda}})}{p(O_1^T|\hat{\mathbf{\Lambda}})} \tag{2.29}$$

$$= \frac{\alpha_i(t-1)\hat{a}_{ij}b_j(o_t)\beta_j(t)}{\alpha_S(T+1)} \tag{2.30}$$

where $\hat{a}_{ij}$ is the current transition probability.

GMM observation probabilities:

When optimizing GMM observation probabilities, the problem in equation (2.23) becomes maximizing the following function:

$$\mathcal{Q}(\mathbf{\Lambda}^{obs}, \hat{\mathbf{\Lambda}}^{obs}) = K_{obs} + \sum_{t=1}^{T}\sum_{j=1}^{S}\sum_{m=1}^{M} \gamma_{jm}(t) \log b_{jm}(\mathbf{o}_t) \qquad s.t. \sum_{m=1}^{M} c_{jm} = 1 \quad \forall j \tag{2.31}$$

where K_{obs} is a constant subsuming terms independent of the GMM observation probabilities, and

$$b_{jm}(\mathbf{o}_t) = c_{jm}\mathcal{N}(\mathbf{o}_t|\boldsymbol{\mu}_{jm}, \mathbf{\Sigma}_{jm}) \tag{2.32}$$

$$\gamma_{jm}(t) = \frac{p(q_t = j, g_t = m, O_1^T|\hat{\mathbf{\Lambda}})}{p(O_1^T|\hat{\mathbf{\Lambda}})} \tag{2.33}$$

$$= \frac{\alpha_j(t)\beta_j(t)}{\alpha_{\mathcal{N}}(T+1)} p(g_t = m|q_t = j, \hat{\mathbf{\Lambda}}) \tag{2.34}$$

$$= \frac{\alpha_j(t)\beta_j(t)}{\alpha_{\mathcal{N}}(T+1)} \frac{\hat{c}_{jm}\mathcal{N}(\mathbf{o}_t|\hat{\boldsymbol{\mu}}_{jm}, \hat{\mathbf{\Sigma}}_{jm})}{\sum_{m'=1}^{M} \hat{c}_{jm'}\mathcal{N}(\mathbf{o}_t|\hat{\boldsymbol{\mu}}_{jm'}, \hat{\mathbf{\Sigma}}_{jm'})} \tag{2.35}$$

where g_t is the component at t, $\hat{\boldsymbol{\mu}}, \hat{\mathbf{\Sigma}}$ are the current Gaussian means and variance, respectively.

The above two constrained optimizing problems can be easily solved by the **Lagrange Multiplier** method [22] because of the convexity, and a series of update formulae from the closed form solution of the **Lagrange** function can be obtained

$$a_{ij} = \frac{\sum_{t=1}^{T} \gamma_{ij}(t)}{\sum_{t=1}^{T} \sum_{j=1}^{S} \gamma_{ij}(t)} \tag{2.36}$$

$$c_{jm} = \frac{\sum_{t=1} \gamma_{jm}(t)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{jm}(t)} \tag{2.37}$$

$$\boldsymbol{\mu}_{jm} = \frac{\sum_{t=1}^{T} \gamma_{jm}(t) \mathbf{o}_t}{\gamma_{jm}(t)} \tag{2.38}$$

$$\boldsymbol{\Sigma}_{jm} = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)(\mathbf{o}_t - \boldsymbol{\mu}_{jm})(\mathbf{o}_t - \boldsymbol{\mu}_{jm})^T}{\sum_{t=1}^{T} \gamma_{jm}(t)} \tag{2.39}$$

2.2.5 HMM Limitations

So far, acoustic modelling using HMM has been introduced, including the formulation, parameter estimation and Viterbi decoding. Although efficient procedures for parameter estimation and recognition have been developed for HMM, there still exist limitations of using HMM for acoustic modelling. As the mechanism of human speech production is unclear, using HMM as a generative model for speech is simply an approximation. Although HMM assumptions have been widely used for speech recognition, these assumptions are not valid for speech. Speech production is a highly complex human activity, and each word/phone is highly correlated to form a valid and meaningful sentence. Therefore, the observation sequence is highly correlated, which is not consistent with the HMM assumptions. These are the most fundamental limitations of HMM formulation. The usual way to circumvent this problem is trajectory modelling.

Next, although maximum likelihood training approach is simple and efficient, the training objective is not consistent with the recognition objective. In other words, maximum likelihood training does not take the word dependency into consideration. It also does not consider the difference between confused words or phones by recognizer. Since speech is not really produced by HMM, the recognition performance may be hurt. The typical approach to solve this issue is discriminative training.

Third, one most challenging problem for speech recognition is the condition mismatch between training and testing. This problem arises from the fact that there exists many acoustic variations, such as speakers, microphones, channels, environments, etc. Although the acoustic model is statistical, minor change of testing condition may still lead to fatal failure. To solve this problem, adaptation and adaptive training is usually applied.

Fourth, although adaptation techniques can be applied for noise robustness, noise variabilities have more unique challenging problem: speech signal can be corrupted or buried by noises. Even if the training and testing conditions are the same, the recognition

performance can still be very poor if Signal-Noise-Ratio (SNR) is high. Therefore, more effective adaptation techniques needs to be specifically developed for noise robustness.

Fifth, conventional HMM uses GMM to model the state emission probability. Due to the high computation expense, diagonal covariance matrix is used and the observation variable is relatively low dimensional acoustic feature, such as MFCC with up to 2 dynamic parameters. Such emission probability computation has ignored the correlation of both inter-frame and intra-frame correlation. Instead of using GMM, a new Deep Neural Network (DNN) technique has been introduced to provide high quality recognition performance. DNN can be viewed as a combination of trajectory modelling and discriminative training, which has become the most widely used technique recently.

Finally, modern ASR systems require a lot of training data to achieve a robust parameter estimation and wide variation coverage. In typical, data collection is an expensive task in terms of time and money. For those languages with temporary interests, it is probably not a good idea to collect hundreds or thousands hours of data for each language. Considering the similarity of phonemes from different languages, cross-lingual speech recognition may be applied.

In next section, several important techniques related to my research interests will be reviewed with details.

2.3 State-of-the-art Techniques

In previous section, efficient parameter estimation algorithm and Viterbi decoding algorithm have been reviewed. However, conventional HMM system itself has a lot of limitations, which hinders its applications for some circumstances. Although researchers have invented many advanced technologies to solve various speech recognition problems, only a few of them related to this thesis work will be reviewed in this section, including trajectory modelling, discriminative training, adaptation and adaptive training, noise robust speech recognition, Deep Neural Network (DNN) and cross-lingual speech recognition.

2.3.1 Trajectory Modelling

When trajectory is discussed in the speech domain, it is referred to as the shape of speech signal with noise removed. In a statistical view, trajectory can be viewed as the mean sequence of the speech signal or observation sequence. Trajectory demonstrates how the speech varies with time and how these frames are correlated to produce a meaningful speech utterance. Otherwise, if no correlation exists, the signal will become some random noise. Conventional HMM treats speech as segments of stationary signals, represented by the HMM states. The observations within each state are assumed to be independent and identically distributed (i.i.d). Moreover, the observations from different states are also assumed to be independent. These assumptions make the parameter estimation

and decoding very simple and efficient. However, this leads to a poor trajectory model. Since trajectory holds rich temporal context information of speech, it motivates many researchers to work on trajectory modelling, either explicitly or implicitly.

2.3.1.1 Explicit Trajectory Modelling

Explicit trajectory modelling tries to model a smooth trajectory which fits the curve of the speech signal as close as possible. One typical approach called parametric trajectory model [23, 24] tried to model each speech segment as a curve rather than a constant trajectory as Figure. 2.6. Low degree polynomials are used for modelling these trajectories. Given a speech segment with length N, a 1-dimension feature can viewed as:

$$c_n = \mu_n + e_n \quad n = 1, \ldots, N, \quad e_n \sim \mathcal{N}(0, \sigma_n) \tag{2.40}$$

where c_n is the feature sequence, μ_n is the mean sequence (a.k.a trajectory), and e_n is the noise term which is assumed to be Gaussian distributed.

What the parametric trajectory model does is to model μ_n as a quadratic function of time, i.e.

$$\mu_n = b_1 + b_2 n + b_3 n^2 \quad n = 1, \ldots, N \tag{2.41}$$

where b_1, b_2, b_3 are the coefficients to learn. In order to do such explicit trajectory modelling, segmentation has to be performed ahead. Therefore, parametric trajectory modelling is also called segmental modelling. However, only small phone classification task was reported in [23, 24]. Generally, segmental models suffer from the inefficient decoding algorithm, and lead to slow progress in LVCSR task.

Recent trajectory modelling for speech recognition using the relationship between static and dynamic features has been investigated. This relationship comes from the fact that the dynamic features are a function of the static features, which is ignored by the conventional ASR systems. Given a sequence of static features, $\{c_1, c_2, \ldots, c_T\}$, the first two order differential features may be obtained using

$$\Delta c_t = \frac{\sum_{i=1}^{\delta} i(c_{t+i} - c_{t-i})}{2\sum_{i=1}^{\delta} i^2} \tag{2.42}$$

and

$$\Delta^2 c_t = \frac{\sum_{i=1}^{\delta} i(\Delta c_{t+i} - \Delta c_{t-i})}{2\sum_{i=1}^{\delta} i^2} \tag{2.43}$$

A recognition method that generates a speech trajectory using an HMM-based speech synthesis method is proposed in [25, 26, 27]. These methods use the standard HMM to generate top three candidates, the HMM-based speech synthesized trajectories are then

generated, finally use the following equation to calculate the likelihood and reorder the candidates.

$$\begin{aligned} &P(C, \Delta C, \Delta^2 C, S|O, \Delta O, \Delta^2 O, \Sigma', \Delta\Sigma', \Delta^2\Sigma') \\ &= \prod_{t=1}^{T} a_{t,t+1} \prod_{t=1}^{T} p(c_t|o_t, \Sigma'_t) p(\Delta c_t|\Delta o_t, \Delta\Sigma'_t) p(\Delta^2 c_t|\Delta^2 o_t, \Delta^2\Sigma'_t) \end{aligned} \tag{2.44}$$

where $\{C, \Delta C, \Delta^2 C\}$ are the static and dynamic parameters, $\{O, \Delta O, \Delta^2 O\}$ are the synthesized trajectories used for taking place of the HMM mean sequence, $\{\Sigma', \Delta\Sigma', \Delta^2\Sigma'\}$ are the recalculated variance for the synthesized trajectories, and S is the state sequence from the Viterbi alignment. This work was tested for word recognition and showed promising gain. Additionally, the re-scoring based technique showed efficient recognition since the likelihood function can be calculated frame by frame.

However, the work in [25, 26, 27] is not a real trajectory model but take advantage of the discriminative power of trajectories, which motivates the researchers to explore more trajectory models based on such relations. One typical example is trajectory HMM: a new kind of trajectory modelling based on HMM with the explicit relationship between static and dynamic features [28, 29, 30, 31], which have been shown some promise gain for phoneme recognition. Current implementation of trajectory HMM can only generate a sub-optimal state sequence because of the intractable best state sequence. The likelihood function of trajectory HMM is formulated as a joint probability of static parameters $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_T^T]^T$ and a state sequence $\mathbf{q}$ given the current model $\mathbf{\Lambda}$.

$$\begin{aligned} P(\mathbf{c}, \mathbf{q}|\mathbf{\Lambda}) &= p(\mathbf{c}|\mathbf{q}, \mathbf{\Lambda}) p(\mathbf{q}|\mathbf{\Lambda}) \\ &= \mathcal{N}(\mathbf{c}|\bar{\mathbf{c}}_q, \mathbf{P}_q) \prod_{t=1}^{T} a_{t,t+1} \end{aligned} \tag{2.45}$$

where $\bar{\mathbf{c}}_q$ is the trajectory and $\mathbf{P}_q$ is the covariance matrix of this sentence. This likelihood function (2.45) is more expensive than (2.44), since the covariance matrix $\mathbf{P}_q$ including all the temporal (inter-frame) correlations is almost full and no frame by frame calculation (2.44) can be applied. The state sequence of trajectory HMM during training is no longer the same as the one from the conventional HMM used in [25]. However, for simplicity, the state sequence for re-scoring during recognition can use the conventional HMM state sequence as a approximation. Of course, the state sequence for re-scoring can also be adjusted by the delayed decision Viterbi algorithm [31].

Although trajectory HMM have similar trajectory based re-scoring algorithm and trajectory generation algorithm to approaches in [25, 26, 27], trajectory HMM tried to implement a valid HMM probability model constrained by the relationship between static and dynamic features. Trajectory HMM have exactly the same set of parameters as the conventional HMM, but a different way to train those parameters. This is one difference to work in [25, 26, 27], which still used the conventional HMM to calculate the trajectory.

Another big difference is that trajectory HMM implemented a almost full covariance matrix $\mathbf{P}_q$, while very simple variance of (2.44) is defined.

In summary, in order to achieve the recognition task by smooth trajectory modelling, e.g. parametric trajectory modelling, or trajectory HMM, the state sequence has to be known firstly. In this case, only re-scoring scheme based algorithm can be applied, which may prematurely prune away many hypotheses and is also not practically for LVCSR applications.

2.3.1.2 Implicit Trajectory Modelling

Literature that aims to model a smooth trajectory have not delivered state-of-the-art results. This is probably because a smoother trajectory does not necessarily correspond to the better recognition accuracy. As the key of trajectory modelling is to utilize the rich temporal context information to achieve a better recognition accuracy, various implicit trajectory models have been proposed by using a state emission probability with higher dependency. Explicit time correlation model [32] assumes that the state mean depends on its previous state and previous observation:

$$\boldsymbol{\mu}_{j_t} = \boldsymbol{\mu}_j + \boldsymbol{\Sigma}_{j_t j_{t-1}} \boldsymbol{\Sigma}^{-1}_{j_{t-1} j_{t-1}} (\mathbf{o}_{t-1} - \boldsymbol{\mu}_{j_{t-1}}) \tag{2.46}$$

where $\boldsymbol{\mu}_{j_t}$ is the dynamic mean of state j at time t, $\boldsymbol{\mu}_j$ is the static mean of state j and $\boldsymbol{\Sigma}_{j_t j_{t-1}}$ is the cross covariance matrix.

Instead of simply using single previous observation and state, more dependent observations were introduced by Vector Linear Prediction (VLP) [33]. In this approach, multiple predictors, which is actually a set of affine transformations, are defined to model different correlations from different surrounding observations:

$$\boldsymbol{\mu}_{j_t} = \boldsymbol{\mu}_j + \sum_{p}^{P} \mathbf{A}^p_j (\mathbf{o}_{t+\tau_p} - \boldsymbol{\mu}_{q_{t+\tau_p}}) \tag{2.47}$$

where p is the index of predictors, τ_p is the offset associated with the p^{th} predictor, which can be positive or negative for succeeding or preceding observations, and $\boldsymbol{\mu}_{q_{t+\tau_p}}$ is the static mean of state at time $t + \tau_p$.

A more general higher order HMM [18, 19, 20] has also been proposed by adding more dependent states such as $p(\mathbf{o}_t | j_t, j_{t-1} \dots j_{t-D})$ and $p(j_t | j_{t-1} \dots j_{t-D})$, where D is the order of higher dependency. However, this approach has significantly increased the system complexity and decreased the decoding speed. Furthermore, no state-of-the-art LVCSR results have been reported using above approaches but simple digit/alphabet recognition tasks were evaluated.

Previous examples have tried to add the higher dependency directly on the observation or state. Another group of studies have introduced additional latent variables to build the nonlinear connection between the temporal information and the state distribution.

In [34], an additional hidden variable $\mathbf{h}_t$ is defined to hold those observations that shares the maximum mutual information with the current observation. In order to use more Gaussian components, [34] defines another latent variable v to denote the class of $\mathbf{h}_t$, and finally the state emission probability is formulated as follows:

$$p(\mathbf{o}_t|\mathbf{\Lambda}_t) = \sum_{m=1}^{M}\sum_{v=1}^{V} P(m|j,v)P(v|\mathbf{h}_t)\mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_{jmvt},\mathbf{\Sigma}_{jmv}) \tag{2.48}$$

where

$$\boldsymbol{\mu}_{jmvt} = \mathbf{A}_{jmv}\mathbf{h}_t + \mathbf{b}_{jmv} \tag{2.49}$$

Switching linear dynamical system (SLDS) [35] defines a different state space compared to the previous HMM based models. In SLDS, except the conventional discrete state, a hidden continuous state variable, $\mathbf{v_t}$, is also defined. Therefore, the underlying process can be formulated as follows:

$$\mathbf{v}_t = \mathbf{A}_j\mathbf{v}_{t-1} + \mathbf{b}_j \tag{2.50}$$
$$\mathbf{o}_t = \mathbf{C}_j\mathbf{v}_t + \mathbf{d}_j \tag{2.51}$$

where

$$\mathbf{b}_j \sim \mathcal{N}(\mathbf{b}_j;\boldsymbol{\mu}_{jv},\mathbf{\Sigma}_{jv}) \tag{2.52}$$
$$\mathbf{d}_j \sim \mathcal{N}(\mathbf{d}_j;\boldsymbol{\mu}_{jo},\mathbf{\Sigma}_{jo}) \tag{2.53}$$

After performing some linear algebra, the above transforms can be jointly applied to calculate the dynamic mean at time t:

$$\boldsymbol{\mu}_{j_t} = \mathbf{C}_j(\mathbf{A}_j\mathbf{v}_{t-1} + \boldsymbol{\mu}_{jv}) + \boldsymbol{\mu}_{jo} \tag{2.54}$$

With a complete review of above models, [36] formulated all these approaches using a single *semi-parametric* trajectory model framework as follows:

$$\boldsymbol{\mu}_{jmt} = \mathbf{A}_t\boldsymbol{\mu}_{jm} + \mathbf{b}_t \tag{2.55}$$
$$\mathbf{P}_{jmt} = \mathbf{Z}_t\mathbf{P}_{jm}\mathbf{Z}_t^T \tag{2.56}$$

where $\mathbf{A}_t$ and $\mathbf{Z}_t$ are time-varying linear transformations, and precision matrix, $\mathbf{P}_{jm}$, is the inverse of the covariance matrix, i.e. $\mathbf{P}_{jm} = \mathbf{\Sigma}_{jm}^{-1}$. Particularly, two trajectory models: fMPE [16] and pMPE[37], have been proposed and delivered state-of-the-art performance for various large scale problems. Due to the success of those two models, this thesis will pay more attention on implicit trajectory modelling.

2.3.2 Discriminative Training

Since HMM is a typical parametric model, it is always important to get best parameter estimation for the model. Depending on the property of objective functions, different local or global optimal solutions may be obtained for the same model. In speech recognition, the fundamental objective is that the correct sentence should have the highest probability among all hypotheses, which can be expressed to maximize the following function:

$$\mathcal{F}(\Lambda) = \sum_{r=1}^{R} P(s_r|O_r, \Lambda) \tag{2.57}$$

$$= \sum_{r=1}^{R} \frac{P(O_r|s_r, \Lambda)P(s_r|\Lambda)}{P(O_r|\Lambda)} \tag{2.58}$$

$$= \sum_{r=1}^{R} \frac{P(O_r|s_r, \Lambda)P(s_r|\Lambda)}{\sum_u P(O_r|u, \Lambda)P(u|\Lambda)} \tag{2.59}$$

where s_r is the r-th training utterance, u represents all possible hypotheses. Maximum Likelihood estimation (MLE) [38] actually assumes that all the hypothesized sentences and acoustic data are uniformly distributed. Mathematically, the objective function of MLE is expressed as:

$$\mathcal{F}_{ML}(\Lambda) = \sum_{r=1}^{R} \log P(O_r|s_r, \Lambda) \tag{2.60}$$

As only the correct hypothesis is considered during ML training, efficient training scheme based on Expectation-Maximisation (EM) estimation [3] has been invented and widely used. Alternatively, discriminative training may also be performed without the above assumptions. For example, Maximum Mutual Information (MMI) [39] has been successfully proposed for speech recognition:

$$\mathcal{F}_{MMI}(\Lambda) = \sum_{r=1}^{R} \log \frac{P(O_r|s_r, \Lambda)^\kappa P(s_r|\Lambda)^\kappa}{\sum_u P(O_r|u, \Lambda)^\kappa P(u|\Lambda)^\kappa} \tag{2.61}$$

$$= \sum_{r=1}^{R} \log \frac{P(O_r|s_r, \Lambda)^\kappa P(s_r|\Lambda)^\kappa}{P(O_r|\Lambda)^\kappa} \tag{2.62}$$

where κ is the probability scale to adjust the importance between language model and acoustic model. Considering that the number of the hypotheses can be huge, a more compact representation of hypotheses, lattice framework [40], has been widely used. Due to the much complicated objective function, discriminative training criteria are much harder to optimize than MLE. Two kinds of techniques are usually employed in practice: 1) gradient based update [41]; 2) the Extended Baum-Welch (EB) update [42, 43].

Another popular training criterion for speech recognition is Minimum Phone Error (MPE) [44] , which aims to minimize (maximize) the phone error rate (accuracy):

$$\mathcal{F}_{MPE}(\lambda) = \sum_{r=1}^{R} \log \sum_{s} \frac{P(O_r|s,\Lambda)^{\kappa} P(s|\Lambda)^{\kappa} A(s,s_r)}{\sum_u P(O_r|u,\Lambda)^{\kappa} P(u|\Lambda)^{\kappa}} \tag{2.63}$$

$$= \sum_{r=1}^{R} \log \sum_{s} P(s|O_r,\Lambda)^{\kappa} A(s,s_r) \tag{2.64}$$

where $A(s, s_r)$ is a function to calculate the "raw phone accuracy" of hypothesis s given the reference transcription s_r. In practice, hypotheses, s are obtained by performing recognition on the training utterance using a ML trained model and a simple language model, such as unigram language model. The resultant hypotheses may be stored as N-best list or in a lattice format, while the latter is usually preferred due to its compact representation. The phone accuracy, $A(s, s_r)$ ideally equals the number of correct phones minus the insertions of the hypothesis compared with the the reference. If the hypotheses are stored in the lattice format, $A(s, s_r)$ can be calculated as the sum of the phone accuracy, $A(q)$, over all phone arcs, q of utterance s in the lattice, where:

$$A(q) = \begin{cases} 1 & \text{if } q \text{ is the correct phone} \\ 0 & \text{if } q \text{ is a substitution/deletion} \\ -1 & \text{if } q \text{ is a insertion} \end{cases} \tag{2.65}$$

As the full alignment of the hypothesis and reference phone sequence is required, exact calculation of phone accuracy can be expensive. Therefore, an approximation was introduced [44] such as:

$$A(q) = \max_{z} \begin{cases} -1 + 2e(q,z) & \text{if } q \text{ and } z \text{ are the same phone} \\ -1 + e(q,z) & \text{otherwise} \end{cases} \tag{2.66}$$

where $e(q, z)$ is the overlapping proportion of q in the duration of reference phone z. If the alignment is perfect, i.e. $e(q, z) = \{0, 1\}$, $A(s, s_r)$ will equal to 1, 0 and -1 for a correct phone, substitution/deletion and insertion, respectively.

In order to optimize above discriminative criteria, not only the likelihood of the correct reference should be maximized, but also the likelihood of the incorrect hypothesis should be minimized. The latter requirement is not needed during ML training. It is important to note that discriminative training usually needs more training data than ML so that the decision boundary can be robustly defined. In practice, an interpolation of ML, MMI and MPE is usually applied in order to obtain better generalization. This is usually implemented by I-Smoothing technique. When MPE sufficient statistics are not enough for updating the model, the update formulae may be downgraded to MMI or ML. Typically, a smoothing constant needs to be defined to adjust the importance of different objective functions.

2.3.3 Speaker Adaptation and Adaptive Training

There are many acoustic factors which can cause variabilities in speech recognition, such as speakers, microphones, environments, etc. Mathematically speaking, the acoustic variabilities can cause the problem that the observation sequence (represented by a series of acoustic features) can be very different for the same word sequence. The same speaker can speak differently at different sessions due to his/her state of emotion, sore throat and aging, which is also called intra-speaker variabilities. Different speaker has different speaking styles such that the acoustic features would be different, which is referred to as inter-speaker variabilities. The typical examples contains gender difference (males and females with different fundamental frequencies), region difference (native and non-native speakers). Besides the speaker variabilities, different recording process can also lead to significant different acoustic features or acoustic patterns. Depending on the recording environments (indoor vs. outdoor), the degree of reverberation can be different. As recording devices and locations/distances can vary (such as close-talk microphone, far-field microphone, mobile phone), the recording channel will be different. If noises exist during recording, different signal-to-noise (SNR) ratio has to be considered, including the noise level and types of noise. In this section, more attentions will be paid to the speaker variabilities. Nevertheless, the techniques designed for speaker variabilities can also be applied for other variabilities in most cases.

Due to existing acoustic variabilities, mismatch problem between training and testing arises in natural. First, feature normalization can be applied to reduce the difference caused by acoustic variations can be reduced. In order to normalize the speaker difference, another feature normalization approach has been developed, i.e. Vocal Tract Length Normalization (VTLN). As different speaker has different size of vocal tracts, VTLN can change the center frequencies of the filter bank analysis by warping the frequency axis. Typically, a warping factor between 0.8 and 1.2 is used to stretch/compress the frequency scale so that the features from different speakers fall on a canonical frequency scale. The actual warping factor can be determined by choosing the highest likelihood of supervised transcription or recognized hypothesis. Since the improvement from feature normalization is limited, it is interesting to consider this problem from the model perspective. There are two typical speech recognition systems: speaker independent and dependent. Speaker independent system is estimated by multi-style training using multi-style training data (from various speakers, microphones, environments, etc.). As this system is designed to cope with various acoustic variabilities, it is mainly used for the cases whose testing speaker is unknown. If the testing speaker is known, speaker dependent system can be trained using only the data collected from a specific speaker. This system aims to cope with intra-speaker variabilities. Providing sufficient training data, speaker dependent system can outperform speaker independent system, since it does not involve the inter-speaker variabilities.

2.3.3.1 Speaker Adaptation

In practice, it is expensive to collect hundreds of hours of training data to estimate a robust speaker dependent speech recognition system. On the other hand, it is much easier to collect the same amount of training data from multiple speakers. Given limited speaker dependent data, robust estimate of speaker dependent system becomes challenging. Various adaptation techniques have been developed to solve this problem [45, 46, 47, 48]. Adaptation aims to update the parameters of a speaker independent system using a small amount of adaptation data such that the system performance can improve towards a speaker dependent system. Depending on the availability of adaptation data, either supervised or unsupervised adaptation can be performed. In case of unsupervised adaptation, recognition by speaker independent system has to be performed first to generate the hypothesis as the supervised data. The performance of unsupervised adaption is largely dependent on the recognition accuracy. If recognition performance is poor, the unsupervised system may become worse. Considering the fact that the adaptation data is limited, it is impossible to directly update all model parameters, which can lead to the over-fitting issue. Two commonly used adaptation approaches are:

Parameter interpolation One simple way is to interpolate the HMM parameters between speaker independent and dependent systems. When more adaptation data is available, the weight on the speaker dependent system increases. Maximum a Posterior (MAP) [45] is just an example of this approach. Alternatively, interpolation of gender dependent systems has also been developed for speaker adaptation [49]. In general, an objective function of MAP estimation may be given as:

$$\Lambda^{MAP} = \arg\max_{\Lambda} p(\Lambda|O_1^T) = \arg\max_{\Lambda} p(O_1^T|\Lambda)p(\Lambda) \tag{2.67}$$

Hence, such adaptation process is also referred to as Bayesian adaptation. This procedure tells the system to use the priors as the model parameters if no adaptation data is observed. In other words, if limited adaptation data is given, a decent MAP estimate may be obtained. Typically, the priors are obtained from the speaker independent model parameters. In order to obtain a tractable optimal solution, conjugate priors are usually used. For example, given the data variance σ^2, in order to estimate MAP mean, assuming that $p(o_t|\mu,\sigma^2) = \mathcal{N}(o_t|\mu,\sigma^2)$ and $p(\mu) = \mathcal{N}(\mu|\mu_0,\sigma_0^2)$, where μ_0 and σ_0 describe a Normal prior distribution of mean, and considering

$$\arg\max_{\mu} p(\mu|O_1^T,\sigma^2) = \arg\max_{\mu} p(O_1^T,\mu|\sigma^2) \tag{2.68}$$

then the problem becomes maximizing the following function:

$$\log p(O_1^T, \mu|\sigma^2) = \sum_{t=1}^{T} \log p(o_t, \mu|\sigma^2) = \sum_{t=1}^{T} \tag{2.69}$$

$$= K - \frac{1}{2}\sum_{t=1}^{T}\left\{\frac{(o_t-\mu)^2}{\sigma^2} + \frac{(\mu-\mu_0)^2}{\sigma_0^2}\right\} \tag{2.70}$$

$$= K' - \frac{1}{2}\frac{(\sigma^2+\sigma_0^2)}{\sigma^2\sigma_0^2}\sum_{t=1}^{T}\left(\mu - \frac{\sigma_0^2 o_t + \sigma^2\mu_0}{\sigma^2+\sigma_0^2}\right) \tag{2.71}$$

Hence, differentiating this function with respect to μ and equating to zero can lead to the optimal solution:

$$\mu^{MAP} = \frac{1}{T}\sum_{t=1}^{T}\frac{\sigma_0^2 o_t + \sigma^2\mu_0}{\sigma^2+\sigma_0^2} \tag{2.72}$$

$$= \frac{\sigma_0^2}{\sigma^2+\sigma_0^2}\mu^{ML} + \frac{\sigma^2}{\sigma^2+\sigma_0^2}\mu_0 \tag{2.73}$$

As can be seen, the MAP estimate is actually an interpolation of maximum likelihood estimate and prior distribution. In practice, as the data variance is usually unknown for speech recognition, the conjugate priors for MAP mean/variance estimates can be normal-Wishart distribution. Hence, the mean update formula for state j and mixture component m becomes

$$\boldsymbol{\mu}_{jm}^{MAP} = \frac{N_{jm}}{N_{jm}+\tau}\boldsymbol{\mu}_{jm}^{ML} + \frac{\tau}{N_{jm}+\tau}\boldsymbol{\mu}_{jm}^{Prior} \tag{2.74}$$

where τ is a weighting of the a priori knowledge to the adaptation speech data and N_{jm} is the occupation of the adaptation data. If more adaptation data are available, more weights will be placed on the maximum likelihood estimation; and vice versa.

Linear regression This approach assumes that the relationship between speaker dependent and independent models is linear. As only regression parameters are updated, the degree of freedom of optimization process can be effectively reduced to avoid the over-training problem. The famous example is Maximum Likelihood Linear Regression (MLLR) [48]. MLLR assumes that the relationship between speaker independent and dependent feature is linear, i.e. $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$, where $\mathbf{x}$ and $\mathbf{y}$ are speaker independent and dependent features respectively. The optimum $\mathbf{A}$ and $\mathbf{b}$ may be found by minimizing the squared error. However, it is not straightforward to prepare the parallel training data. Alternatively, maximum likelihood (ML) estimation can be applied:

$$\mathcal{L}(\Lambda^{MLLR}) = \log p(O_1^T|\Lambda^{MLLR}) = \sum_{1}^{T}\log p(o_t|\Lambda^{MLLR}) \tag{2.75}$$

depending on the actual adaptation methods, different adaptation parameters can be formulated:

$$\text{MLLR for mean:} \quad \boldsymbol{\mu}^{MLLR} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b} \tag{2.76}$$

$$\text{MLLR for variance :} \quad \boldsymbol{\Sigma}^{MLLR} = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T \tag{2.77}$$

$$\text{Constrained MLLR (CMLLR) :} \tag{2.78}$$

$$\boldsymbol{\mu}^{CMLLR} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b} \quad \text{and} \quad \boldsymbol{\Sigma}^{CMLLR} = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T \tag{2.79}$$

where CMLLR [46] uses the same MLLR transform, $\mathbf{A}$ for both mean and variances adaptations. In case of speech recognition system with hundreds of thousands of Gaussian components, it is not possible to estimate a separate transform for each component. In practice, one transform is estimated for one regression class, i.e. a group of Gaussian components. In an extreme case, one transform may be estimated and applied for all components. Therefore, MLLR has more flexibilities of controlling the degree of freedom for speaker dependent system. This is one advantage of MLLR compared to MAP if the adaptation data is relatively small. MAP, however, usually requires more adaptation data for an effective estimate, as it performs the estimate for each component independently. Basically, MAP and MLLR families can be applied for general adaptation tasks. For example, they can be used for both speaker and noise adaptations.

2.3.3.2 Speaker Adaptive Training

So far, adaptation techniques have been discussed to improve the recognition performance using limited amount of adaptation data during the recognition stage. The same technique such as MLLR can also be used to normalize the effects of such as speaker during training. In conventional, multi-style training uses training data from various speakers, whose features lie in different acoustic space. MLLR can transform the features from different speakers into a canonical space, so that a canonical acoustic model can be estimated. This is well known as speaker adaptive training (SAT). During adaptive training, MLLR parameters are estimated for each speaker in the training data. Then, the canonical model parameters (Gaussian parameters and transition probabilities) are estimated given the MLLR adapted statistics. This process is performed iteratively till the likelihood converges. This process also needs to be carried out during recognition: MLLR needs to be iteratively revised till convergence.

Mathematically, the objective of adaptive training using MLLR can be expressed as maximizing following function:

$$\mathcal{Q}(\Lambda; \hat{\Lambda}) = \sum_{k=1}^{K} \sum_{t,j,m} \gamma_{jm}^{k}(t) \log P(o_t^k | \Lambda_{jm}^{SAT}, \Lambda_k^{MLLR}) \tag{2.80}$$

where k is the speaker index, Λ^{SAT} are the canonical acoustic model parameters, Λ_k^{MLLR} are the speaker k dependent regression parameters, $\hat{\Lambda}$ are the current model parameters.

Note that in order to apply the MLLR transform, a regression tree (or regression bases) has to be built to look up the associated regression class (i.e. the actual transform) for each component j, m. Hence, the speaker adaptive training algorithm can be described in a more formal fashion:

1. Initialize the canonical acoustic model $\hat{\Lambda}^{SAT}$ by multi-style training, the regression parameters $\hat{\Lambda}_k^{MLLR}$ with the identity transform and zero bias.

2. Estimate speaker dependent regression parameters Λ^{MLLR} by maximizing auxiliary function $\mathcal{Q}(\Lambda^{MLLR}; \hat{\Lambda}^{MLLR}, \hat{\Lambda}^{SAT})$.

3. Estimate canonical acoustic model parameters Λ^{SAT} by maximizing auxiliary function $\mathcal{Q}(\Lambda^{SAT}; \hat{\Lambda}^{MLLR}, \hat{\Lambda}^{SAT})$

4. If the likelihood does not converge, continue to step 2; otherwise, terminate.

During decoding, a similar process needs to be performed, except that the canonical acoustic model parameters are unchanged. Depending on the tractability of the auxiliary function, other adaptation approaches may also be borrowed for adaptive training.

2.3.4 Noise Robust Speech Recognition

Different from speaker variabilities, noise variabilities have more impact on the system performance. Although Different speaker may have different speaking style, the acoustic pattern will be still there and can be statistically learned with more training data. The difficulty of noise variabilities is that the acoustic pattern may be corrupted or buried by noises if SNR is high and non-stationary noises exist. In that case, even when the training and testing conditions are the same, the recognition performance can still be very poor. Therefore, noise robust speech recognition has become one of the most challenging recognition tasks. Techniques to deal with the noise corrupted speech contains two main categories: features enhancement and model compensation. No clear evidence has been found which one is superior to another. It heavily depends on the available computing resources, i.e. whether it is a response time sensitive or accuracy demanding task. Generally speaking, feature enhancement is more efficient than model compensation, while model compensation is potentially capable of handling more environments and achieving better accuracy.

2.3.4.1 Feature Enhancement

The simplest approach for feature enhancement is cepstral mean/variance normalization. Given an utterance represented by a 1-dimension cepstral feature sequence $\{c_1, \ldots, c_t, \ldots, c_T\}$,

whose mean and variance are μ, σ, respectively, the enhanced feature can be given as:

$$\text{Cepstral Mean Normalization (CMN):} \quad \hat{c}_t = c_t - \mu \tag{2.81}$$

$$\text{Cepstral Variance Normalization (CVN):} \quad \hat{c}_t = \frac{c_t}{\sqrt{\sigma}} \tag{2.82}$$

$$\text{Cepstral Mean\&Variance Normalization (CMVN):} \quad \hat{c}_t = \frac{c_t - \mu}{\sqrt{\sigma}} \tag{2.83}$$

where CMN is actually also an alternative way of removing the channel difference. In practice, if the incoming speech data is long, normalization may be performed per segment. Homogenous segments can be detected by a Voice Activity Detector (VAD). The length of segment is a compromise: longer segment yield better mean and variance, but system has to wait longer till the end of the segment before normalization can be done. These simple approaches are very effective and also show promising improvements over the conventional acoustic features. Spectral noise subtraction was also proposed [50], however the non-speech activity detection is required to estimate the additive noise.

Recently, people are more interested in estimating the clean speech by using minimum mean square error estimation schemes (MMSE) [51, 52, 53, 54]. MMSE may be applied with stereo data [51, 52] or noise model estimation [53, 54]. The idea behind MMSE is that the estimated clean speech $\mathbf{x}$ given the noisy speech $\mathbf{y}$ is the mean of the conditional distribution $p(\mathbf{x}|\mathbf{y})$:

$$\hat{\mathbf{x}} = \mathcal{E}[\mathbf{x}|\mathbf{y}] = \int_{\mathbf{x}} p(\mathbf{x}|\mathbf{y})\mathbf{x}\,\mathrm{d}\mathbf{x} \tag{2.84}$$

$$= \sum_k \int_{\mathbf{x}} p(\mathbf{x}, k|\mathbf{y})\mathbf{x}\,\mathrm{d}\mathbf{x} \tag{2.85}$$

$$= \sum_k P(k|\mathbf{y}) \int_{\mathbf{x}} p(\mathbf{x}|k, \mathbf{y})\mathbf{x}\,\mathrm{d}\mathbf{x} \tag{2.86}$$

$$= \sum_k P(k|\mathbf{y})\mathcal{E}[\mathbf{x}|k, \mathbf{y}] \tag{2.87}$$

where k represents a front-end mixture model for the joint probability of $\{\mathbf{x}, \mathbf{y}\}$ such as

$$p\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\right) = \sum_{k=1}^{K} P(k)\mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_{x,k}\\ \boldsymbol{\mu}_{y,k}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{xx,k} & \boldsymbol{\Sigma}_{xy,k}\\ \boldsymbol{\Sigma}_{yx,k} & \boldsymbol{\Sigma}_{yy,k}\end{bmatrix}\right) \tag{2.88}$$

Therefore,

$$\mathcal{E}[\mathbf{x}|k, \mathbf{y}] = \boldsymbol{\mu}_{x,k} + \boldsymbol{\Sigma}_{xy,k}\boldsymbol{\Sigma}_{yy,k}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{y,k}) \tag{2.89}$$

and the posterior can be obtained as

$$P(k|\mathbf{y}) = \frac{p(\mathbf{y}|k)P(k)}{\sum_k p(\mathbf{y}|k)P(k)} \tag{2.90}$$

In some cases, fast estimation of clean speech can be obtained by using just single most likely mixture:

$$\hat{\mathbf{x}} = \mathcal{E}[\mathbf{x}|\hat{k}, \mathbf{y}], \qquad \hat{k} = \arg\max_k P(k|\mathbf{y}) \tag{2.91}$$

Hence, the likelihood of noisy speech is approximated by passing the estimated clean speech to the back-end clean recognizer:

$$p(\mathbf{y}|m) \approx \mathcal{N}(\hat{\mathbf{x}}; \boldsymbol{\mu}_{x,m}, \boldsymbol{\Sigma}_{x,m}) \tag{2.92}$$

However, this approach has difficulties to get an accurate estimation of clean speech when the Signal-Noise-Ratio (SNR) is very low where $\mathbf{y} \approx \mathbf{n}$, $\mathbf{n}$ is the noise. [55, 56] proposed an uncertainty decoding scheme by marginalizing over the clean speech to obtain the likelihood such as

$$p(\mathbf{y}|m) = \int p(\mathbf{y}|x, m)p(\mathbf{x}|m)\,\mathrm{d}\mathbf{x} \approx \int p(\mathbf{y}|x)p(\mathbf{x}|m)\,\mathrm{d}\mathbf{x} \tag{2.93}$$

where

$$p(\mathbf{y}|\mathbf{x}) = \sum_k P(k|\mathbf{x})p(\mathbf{y}|\mathbf{x}, k) \tag{2.94}$$

However, the posterior $P(k|\mathbf{x})$ is very difficult to calculate, since $\mathbf{x}$ is unknown. Approximations are made in [55, 56].

2.3.4.2 Model Compensation

Similar to other speech related variations, general adaptation techniques may be directly applied to handle the noisy environments, such as Maximum Likelihood Linear Regression (MLLR) [48], Constrained MLLR [46], and Maximum a Posterior (MAP) [45]. However, these approaches are highly subject to the amount of available adaptation data. In addition, MLLR or CMLLR transform is only linear, or piece-wise linear, while the impact of noise on speech is highly nonlinear. In order to obtain a better estimation of noise condition dependent models, various model compensation approaches were proposed, such as Parallel Model Combination (PMC) [57], Vector Taylor Series (VTS) [58], and many other derivatives Extended VTS [59], Data-driven PMC [60], Trajectory-based PMC [61]. All these model compensation approaches have one common requirement: noise model as a representation of the noisy environment.

First, a typical model of the acoustical environment is introduced as in Figure. 2.8. This model assumes the speech signal $x[m]$ is corrupted by an additive noise $n[m]$ and channel distortion $h[m]$. Mathematically speaking, corrupted speech signal $y[m]$ can be written as a linear function of these quantities:

$$y[m] = x[m] * h[m] + n[m] \tag{2.95}$$

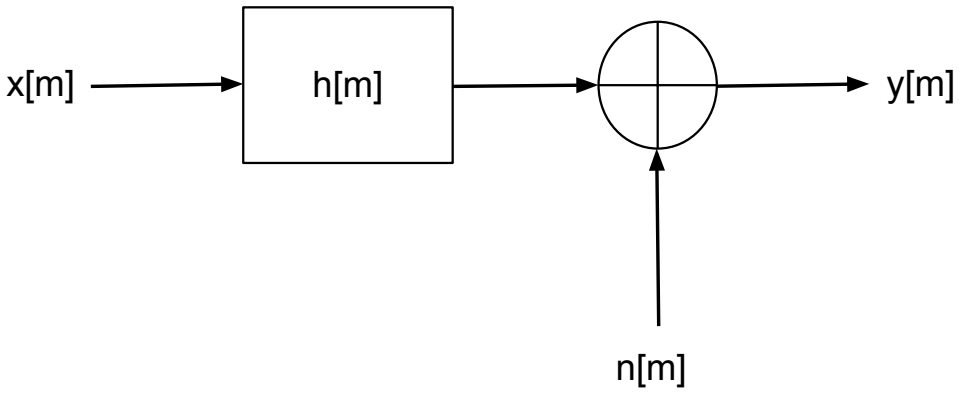

Figure 2.8: A typical model of the acoustical environment

where m indicates the signal sample index. After taking the square of the complex variables from a Q-point Discrete Fourier Transform (DFT), the spectral magnitudes relationship becomes

$$|Y(q_i)|^2 = |X(q_i)|^2|H(q_i)|^2 + |N(q_i)|^2 + 2Real\left\{X(q_i)H(q_i)N(q_i)\right\} \tag{2.96}$$

where $X(q_i), Y(q_i), H(q_i), N(q_i)$ represent the frequency analysis of clean speech, noisy speech, channel distortion and additive noise, respectively, operator $Real\{\}$ extracts the real component of the complex variable, $i = 0, 1, \ldots, Q$. Since $x[m]$ and $n[m]$ are statistically independent, the last term, $Real\left\{X(q_k)H(q_k)N(q_k)\right\}$ is expected to be zero. However, it is not zero in practice [58], particular when computing the famous mel-cepstrum, which requires a range of frequencies. Anyway, the spectrum relationship by applying M (mel) filter banks is assumed to be:

$$|Y(f_i)|^2 = |X(f_i)|^2|H(f_i)|^2 + |N(f_i)|^2 \tag{2.97}$$

where $i = 0, 1, \ldots, M$. Note that $|Y(f_i)|^2$ is the sum of ith-bandpassed spectral magnitudes from a range of frequencies, $|Y(q_.)|^2$.

After taking the logarithms in Eq-2.97 and some algebraic manipulation, the log-spectrum relationship can be obtained as:

$$\log|Y(f_i)|^2 = \log|X(f_i)|^2 + \log|H(f_i)|^2 \tag{2.98}$$

$$+ \log\left(1 + \exp\left\{\log|N(f_i)|^2 - \log|X(f_i)|^2 - \log|H(f_i)|^2\right\}\right) \tag{2.99}$$

Then, the cepstrum features for each signal can be obtained by taking Discrete Cosine Transform (DCT), such as

$$\mathbf{x} = \mathbf{C}\left(\log(|X(f_0)|^2) \quad \log(|X(f_1)|^2) \quad \cdots \quad \log(|X(f_M)|^2)\right) \tag{2.100}$$
$$\mathbf{y} = \mathbf{C}\left(\log(|Y(f_0)|^2) \quad \log(|Y(f_1)|^2) \quad \cdots \quad \log(|Y(f_M)|^2)\right) \tag{2.101}$$
$$\mathbf{h} = \mathbf{C}\left(\log(|H(f_0)|^2) \quad \log(|H(f_1)|^2) \quad \cdots \quad \log(|H(f_M)|^2)\right) \tag{2.102}$$
$$\mathbf{n} = \mathbf{C}\left(\log(|N(f_0)|^2) \quad \log(|N(f_1)|^2) \quad \cdots \quad \log(|N(f_M)|^2)\right) \tag{2.103}$$

Therefore, the cepstrum relationship can be given as

$$\mathbf{y} = \mathbf{x} + \mathbf{h} + \mathbf{g}\left(\mathbf{n} - \mathbf{x} - \mathbf{h}\right) \tag{2.104}$$

where $\mathbf{g}(\mathbf{z})$ is a nonlinear function, given as

$$\mathbf{g}(\mathbf{z}) = \mathbf{C}\log\left(1 + \exp\{\mathbf{C}^{-1}\mathbf{z}\}\right) \tag{2.105}$$

where $\mathbf{C}^{-1}$ is the pseudo-inverse matrix of DCT matrix $\mathbf{C}$, since $\mathbf{C}$ used in feature extraction is not square. The function in Eq-2.104 plays a very important role in the most popular VTS model compensation approach [58], since the noise corruption process is directly expressed in terms of the widely used cepstrum features.

Now, we can discuss how the most popular VTS was proposed for model compensation. Although the noisy speech can be obtained according to Eq-2.104, statistics of noisy speech, $\{\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y\}$ are difficult to calculate due to the nonlinearity of function $\mathbf{g}(\mathbf{z})$. Therefore, Eq-2.104 may be approximated by first order Vector Taylor Series at $\{\mathbf{x}_0, \mathbf{h}_0, \mathbf{n}_0\}$:

$$\hat{\mathbf{y}} \approx \mathbf{x}_0 + \mathbf{h}_0 + \mathbf{g}_0 + \left.\frac{\partial \mathbf{y}}{\partial \mathbf{n}}\right|_{\mathbf{n}=\mathbf{n}_0} + \left.\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} + \left.\frac{\partial \mathbf{y}}{\partial \mathbf{h}}\right|_{\mathbf{h}=\mathbf{h}_0} \tag{2.106}$$

where

$$\mathbf{g}_0 = \mathbf{g}\left(\mathbf{n}_0 - \mathbf{x}_0 - \mathbf{h}_0\right) \tag{2.107}$$
$$\left.\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} = \left.\frac{\partial \mathbf{y}}{\partial \mathbf{h}}\right|_{\mathbf{h}=\mathbf{h}_0} = \mathbf{G} = \mathbf{C}\,\mathrm{Diag}\left(\frac{1}{1 + \exp\{\mathbf{C}^{-1}(\mathbf{n}_0 - \mathbf{x}_0 - \mathbf{h}_0)\}}\right)\mathbf{C}^{-1} \tag{2.108}$$
$$\left.\frac{\partial \mathbf{y}}{\partial \mathbf{n}}\right|_{\mathbf{n}=\mathbf{n}_0} = \mathbf{F} = \mathbf{I} - \mathbf{G} \tag{2.109}$$

where the operator $\mathrm{Diag}(\mathbf{v})$ builds a diagonal matrix whose principle diagonal element is given by vector $\mathbf{v}$.

Now, we define $\{\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x\}$,$\{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\}$ and $\{\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h\}$ as the statistics of $\mathbf{x}, \mathbf{n}, \mathbf{h}$ respectively. After setting the expansion point at $\{\mathbf{x}_0, \mathbf{h}_0, \mathbf{n}_0\} = \{\boldsymbol{\mu}_x, \boldsymbol{\mu}_h, \boldsymbol{\mu}_n\}$, we can have following estimation of corrupted speech statistics:

$$\hat{\mathbf{y}} \approx \boldsymbol{\mu}_x + \boldsymbol{\mu}_h + g(\boldsymbol{\mu}_n - \boldsymbol{\mu}_x - \boldsymbol{\mu}_h) + \mathbf{G}(\mathbf{x} - \boldsymbol{\mu}_x) + \mathbf{G}(\mathbf{h} - \boldsymbol{\mu}_h) + \mathbf{G}(\mathbf{n} - \boldsymbol{\mu}_n) \tag{2.110}$$
$$\boldsymbol{\mu}_y = \mathcal{E}[\hat{\mathbf{y}}] \approx \boldsymbol{\mu}_x + \boldsymbol{\mu}_h + g(\boldsymbol{\mu}_n - \boldsymbol{\mu}_x - \boldsymbol{\mu}_h) \tag{2.111}$$
$$\boldsymbol{\Sigma}_y = \mathcal{E}[(\hat{\mathbf{y}} - \boldsymbol{\mu}_y)(\hat{\mathbf{y}} - \boldsymbol{\mu}_y)^T] \approx \mathbf{G}\boldsymbol{\Sigma}_x\mathbf{G}^T + \mathbf{G}\boldsymbol{\Sigma}_h\mathbf{G}^T + \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^T \tag{2.112}$$

In practice, $\mathbf{\Sigma}_x, \mathbf{\Sigma}_n, \mathbf{\Sigma}_h$ are diagonal, however $\mathbf{\Sigma}_y$ is no longer diagonal. An assumption forcing $\mathbf{\Sigma}_y$ to be diagonal is usually made for efficient decoding. In case of computing the statistics of corrupted dynamic parameters, $\{\Delta\boldsymbol{\mu}_y, \Delta\mathbf{\Sigma}_y\}$ and $\{\Delta^2\boldsymbol{\mu}_y, \Delta^2\mathbf{\Sigma}_y\}$, continuous-time approximation is proposed [62] Given the facts by continuous-time approximation:

$$\Delta\mathbf{x}_t = \mathbf{x}_{t+2} - \mathbf{x}_{t-2} \tag{2.113}$$

$$\Delta\mathbf{x} = 4\frac{\partial\mathbf{x}_t}{\partial t} \tag{2.114}$$

we can have

$$\frac{\partial\mathbf{y}}{\partial t} \approx \mathbf{G}\frac{\partial\mathbf{x}}{\partial t} \tag{2.115}$$

so that

$$\boldsymbol{\mu}_{\Delta y} \approx \mathbf{G}\boldsymbol{\mu}_{\Delta x} \tag{2.116}$$

$$\mathbf{\Sigma}_{\Delta y} \approx \mathbf{G}\mathbf{\Sigma}_{\Delta x}\mathbf{G}^T + \mathbf{F}\mathbf{\Sigma}_{\Delta n}\mathbf{F}^T \tag{2.117}$$

where $\mathbf{h}$ is assumed to be constant within the utterance such that $\mathbf{\Delta h} = \mathbf{0}$. Similarly, we can obtain the approximation for the second order dynamic parameters' statistics such as:

$$\boldsymbol{\mu}_{\Delta^2 y} \approx \mathbf{G}\boldsymbol{\mu}_{\Delta^2 x} \tag{2.118}$$

$$\mathbf{\Sigma}_{\Delta^2 y} \approx \mathbf{G}\mathbf{\Sigma}_{\Delta^2 x}\mathbf{G}^T + \mathbf{F}\mathbf{\Sigma}_{\Delta^2 n}\mathbf{F}^T \tag{2.119}$$

Although VTS compensation itself looks quite elegant, there still remains one unsolved problem: the noise model, $\{\boldsymbol{\mu}_n, \mathbf{\Sigma}_n, \boldsymbol{\mu}_h, \mathbf{\Sigma}_h\}$ is unknown. This problem is to be introduced and addressed in later chapter, and so is noise adaptive training (NAT) [63].

2.3.5 Deep Neural Network (DNN)

Recently, Deep Neural Network (DNN) [64, 65] has been successfully applied for LVCSR tasks and shown dramatic performance improvements over GMM based HMM systems. One main contribution is that pre-training scheme of Deep Belief Network (DBN) using modern GPU's vector processing capability is introduced as initialization of DNN such that the entire training context dependent (CD) DNN becomes tractable. Some characteristics of DNN that are good for speech recognition can be found. First, DNN can model the long-term contextual information via using long span of acoustic features as input and multiple layers of non-linear processing. However, this is not possible for GMM/HMM systems, as diagonal covariance matrix is assumed for robust estimation and efficient decoding. Literature has also shown that DNN using single frame as input performs worse than the simple GMM system [66]. On the other hand, DNN does not assume the data distribution, which is not necessary for problems like classification. In this section, the framework of CD-DNN/HMM will be briefly introduced, including the Restricted Boltzmann Machine, Pre-training, fine-tuning and decoding.

2.3.5.1 Restricted Boltzmann Machine (RBM)

Firstly, a brief introduction of Restricted Boltzmann Machine (RBM) [67] is given, which serves as the fundamental component of DNN pre-training. RBM is an undirected graphical model constructed from a layer of binary stochastic hidden units and a layer of stochastic visible units, as shown in Figure. 2.9, and it is also a generative stochastic model that can learn a probability distribution over its set of inputs. Note that RBM only has connections between inter-layer units but no connections between intra-layer units, otherwise it will become unrestricted Boltzmann machine. In RBM, an energy

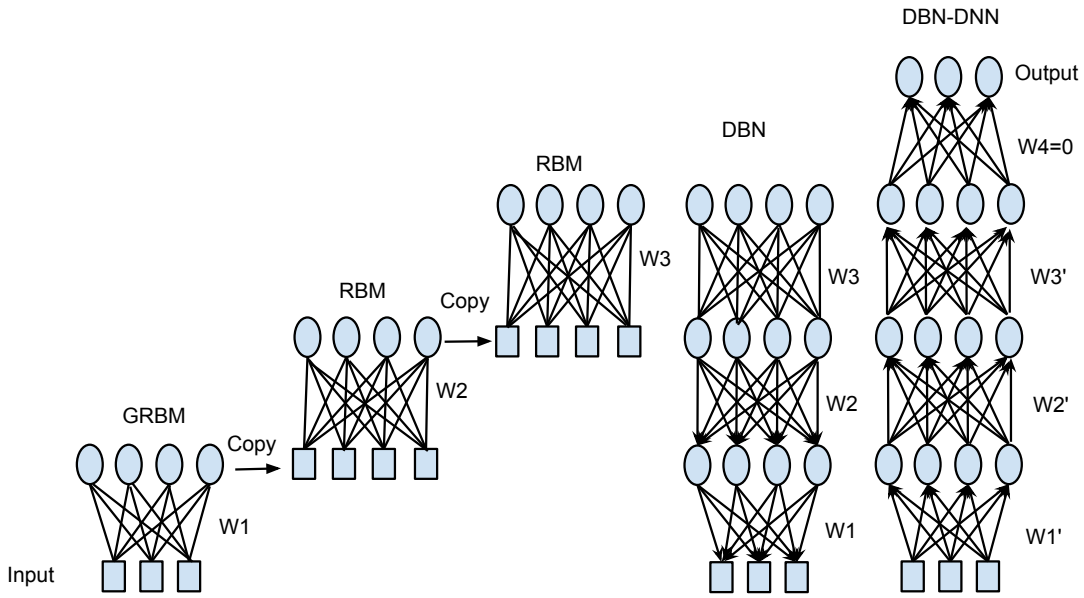

Figure 2.9: A diagram of DBN pre-training process for DNN initialization, where square box represents visible units while oval represents hidden units.

analogous to that of a Hopfield network is used:

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{b}^T\mathbf{v} - \mathbf{c}^T\mathbf{h} - \mathbf{v}^T\mathbf{W}\mathbf{h} \tag{2.120}$$

where $\mathbf{v}$ and $\mathbf{h}$ are the binary vectors for visible and hidden units, respectively, $\mathbf{b}$ is the visible unit bias, $\mathbf{c}$ is the hidden unit bias, $\mathbf{W}$ is the matrix of visible/hidden connection

weights. Hence, the energy term can be used to define the probability of any visible/hidden units such as

$$P(\mathbf{v}, \mathbf{h}) = \frac{\exp\{-E(\mathbf{v}, \mathbf{h})\}}{Z} \tag{2.121}$$

where Z is the normalization term given as

$$Z = \sum_{\mathbf{v},\mathbf{h}} \exp\{-E(\mathbf{v}, \mathbf{h})\} \tag{2.122}$$

After some mathematical derivations [65], we can obtain:

$$P(\mathbf{h}|\mathbf{v}) = \frac{\exp\{-E(\mathbf{v}, \mathbf{h})\}}{\sum_{\tilde{\mathbf{h}}} \exp\{-E(\mathbf{v}, \tilde{\mathbf{h}})\}} = \prod_i P(h_i|\mathbf{v}) \tag{2.123}$$

$$P(\mathbf{h} = \mathbf{1}|\mathbf{v}) = \sigma(\mathbf{c} + \mathbf{v}^T\mathbf{W}) \tag{2.124}$$

$$P(\mathbf{v} = \mathbf{1}|\mathbf{h}) = \sigma(\mathbf{b} + \mathbf{h}^T\mathbf{W}^T) \tag{2.125}$$

where $\mathbf{1}$ is a vector with all elements equating 1, $\sigma(x)$ denotes the logistic sigmoid function such as

$$\sigma(x) = \frac{1}{1 + \exp\{x\}} \tag{2.126}$$

In fact, Eq-2.125 can be viewed as a reconstruction process, which can be used to reconstruct the most likely visible configuration given the hidden units. On the other hand, Eq-2.124 is viewed as an inference process, which can be used to predict the best hidden configuration given the visible units. Due to the same sigmoid activation function, the connection weights of RBM can be then applied to initialize a feed-forward neural network with sigmoid hidden units.

In order to estimate the model parameters, maximum likelihood estimation is applied. Given the probability of a visible unit $\mathbf{v}$:

$$P(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h}) = \frac{\sum_{\mathbf{h}} \exp\{-E(\mathbf{v}, \mathbf{h})\}}{\sum_{\boldsymbol{\nu},\mathbf{h}} \exp\{-E(\boldsymbol{\nu}, \mathbf{h})\}} \tag{2.127}$$

then the log-likelihood can be obtained to estimate the connection weights such as:

$$\mathcal{L}(\boldsymbol{\Lambda}) = \log\left(\sum_{\mathbf{h}} \exp\{-E(\mathbf{v}, \mathbf{h})\}\right) - \log\left(\sum_{\boldsymbol{\nu},\mathbf{h}} \exp\{-E(\boldsymbol{\nu}, \mathbf{h})\}\right) \tag{2.128}$$

where $\boldsymbol{\Lambda}$ are the model parameters. Typically, a quantity known as the free energy is defined as:

$$\mathcal{F}(\mathbf{v}) = -\log\left(\sum_{\mathbf{h}} \exp\{-E(\mathbf{v}, \mathbf{h})\}\right) \tag{2.129}$$

Therefore, the log-likelihood function can be rewritten as:

$$\mathcal{L}(\boldsymbol{\Lambda}) = -\mathcal{F}(\mathbf{v}) - \log\left(\sum_{\boldsymbol{\nu}} \exp\{-\mathcal{F}(\boldsymbol{\nu})\}\right) \tag{2.130}$$

whose negative gradient has a very elegant form:

$$-\frac{\partial \mathcal{L}(\mathbf{\Lambda})}{\partial \mathbf{\Lambda}} = \frac{\partial \mathcal{F}(\mathbf{v})}{\partial \mathbf{\Lambda}} - \sum_{\boldsymbol{\nu}} p(\boldsymbol{\nu}) \frac{\partial \mathcal{F}(\boldsymbol{\nu})}{\partial \mathbf{\Lambda}} \tag{2.131}$$

$$= \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \mathbf{\Lambda}} - \sum_{\boldsymbol{\nu}, \mathbf{h}} P(\boldsymbol{\nu}, \mathbf{h}) \frac{\partial E(\boldsymbol{\nu}, \mathbf{h})}{\partial \mathbf{\Lambda}} \tag{2.132}$$

$$= \left\langle \frac{\partial E}{\partial \mathbf{\Lambda}} \right\rangle_{data} - \left\langle \frac{\partial E}{\partial \mathbf{\Lambda}} \right\rangle_{model} \tag{2.133}$$

where $\langle . \rangle_{data}$ is the expectation over the distribution from the visible data, $\langle . \rangle_{model}$ is the same expectation over the distribution from the reconstructed data by the current model. Stochastic gradient descent method can be applied to minimize the negative log-likelihood using following update direction:

$$w_{ij}(t+1) = w_{ij}(t) - \Delta w_{ij}(t+1) \tag{2.134}$$

$$\Delta w_{ij}(t+1) = m \Delta w_{ij}(t) - \alpha \frac{\partial \mathcal{L}(\mathbf{\Lambda})}{\partial w_{ij}} \tag{2.135}$$

where the gradient of negative log-likelihood function is given as:

$$-\frac{\partial \mathcal{L}(\mathbf{\Lambda})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{2.136}$$

α is the learning rate, m is the "momentum" factor to smooth the weight updates, t is the update iteration, Therefore, maximizing the log-likelihood of the data is exactly the same as minimizing the Kullback-Leibler divergence, $KL(P^0||P^\infty_{\mathbf{\Lambda}})$, where P^0 is the distribution of data, $P^\infty_{\mathbf{\Lambda}}$ is the equilibrium distribution defined by the model. In practice, contrastive divergence learning [68] is adopted by minimizing the following two KL divergence:

$$KL(P^0||P^\infty_{\mathbf{\Lambda}}) - KL(P^n_{\mathbf{\Lambda}}||P^\infty_{\mathbf{\Lambda}}) \tag{2.137}$$

where $P^n_{\mathbf{\Lambda}}$ is the distribution after n step of Gibbs sampling for reconstruction. For a greedy algorithm, n may be set as 1 such that the gradient can be given as:

$$-\frac{\partial \mathcal{L}(\mathbf{\Lambda})}{\partial w_{ij}} \approx \left\{ \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{\infty} \right\} - \left\{ \langle v_i h_j \rangle_{1} - \langle v_i h_j \rangle_{\infty} \right\} \tag{2.138}$$

$$= \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{1} \tag{2.139}$$

This is also called one-step contrastive divergence approximation [65] or CD-1.

Gaussian-Bernoulli RBM (GRBM) Since the acoustic features are real-valued vectors instead of binary vectors, Gaussian-Bernoulli RBM (GRBM) is introduced, whose energy function is given as:

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2}(\mathbf{v} - \mathbf{b})^T(\mathbf{v} - \mathbf{b}) - \mathbf{c}^T\mathbf{h} - \mathbf{v}^T\mathbf{W}\mathbf{h} \tag{2.140}$$

where a diagonal covariance Gaussian noise model with a variance of 1 on each dimension is implicitly assumed. After switching to GRBM, the conditional probabilities of hidden units in Eq-2.124 keep the same, while the conditional probabilities of visible units are revised as

$$P(\mathbf{v}|\mathbf{h}) = \mathcal{N}(\mathbf{v}; \mathbf{b} + \mathbf{h}^T\mathbf{W}^T, \mathbf{I}) \tag{2.141}$$

where $\mathbf{I}$ is an identity matrix. In practice, when training a GRBM and creating a reconstruction, the reconstructed visible units are simply set to be equal to their means rather than sampled from the above posterior probability.

2.3.5.2 DBN Pre-training

In Figure. 2.9 [69], a diagram of DBN pre-training for DNN initialization is shown. Before training RBM, the acoustic features have to be normalized to a canonical space, which can be achieved by cepstral mean and variance normalization. The mean and variance for normalization can be either global or per utterance. In case of normalization per utterance, some other variations such as channel difference for different recording session can also be normalized. Such normalization will also help keep the weight initialization more effective. In previous section, the parameter learning process has been introduced for one RBM with single visible and hidden layers. In order to pre-train a deep belief network (DBN), a greedy algorithm has been designed [70]. Considering the diagram shown in Figure. 2.9, first, given the real-value visible observations, $\mathbf{v}^{(1)}$, $\mathbf{W}_1$ of GRBM is learned by using contrastive divergence algorithm, i.e. CD-1. Then, posterior probabilities of hidden states, $p(\mathbf{h}_i^{(1)}|\mathbf{v}^{(1)})$ can be inferred using the current model, $\mathbf{W}_1$. Those posterior probabilities can serve as the new visible units to train another RBM, $\mathbf{W}_2$. This inference and estimation process will continue till the expected number of RBMs reaches. Once this process completes, multiple RBMs will be stacked together to form a single generative model, a deep belief network (DBN). At the same time, undirected connection of those RBMs will be replaced by top-down directed connection. As the RBM is not a perfect model of the original data, the stacked RBMs is just an approximation of DBN. Despite the approximation, it provides a good initialization of DBN and also saves a lot of computing time. Recent findings also show that the pre-training is less crucial as the training data increases [71]. However, pre-training is still an effective approach for initializing deep neural networks.

2.3.5.3 CD-DNN/HMM Fine-tuning and Decoding

After the deep neural network is initialized by DBN pre-training, supervised fine-tuning has to be performed in order to predict the posterior of context dependent states or any other classification label. First, an output layer with softmax activation function has to

be appended to the top layer of DBN. Given the forward probabilities of top hidden layer, $\mathbf{h}^{(K)}$, the final layer output probability of label l can be formulated as:

$$p(l|\mathbf{h}^{(K)}) = \frac{\exp\{b_l + \sum_i h_i^{(K)} w_{il}\}}{\sum_k \exp\{b_k + \sum_i h_i^{(K)} w_{ik}\}} \tag{2.142}$$

where b_l is the bias of label l, w_{il} is the weight connecting the hidden unit i to output label l. The supervision label for each time frame can be obtained by performing force alignment of a well trained context dependent GMM/HMM system. Hence, the neural network parameter may be optimized by minimizing the cross-entropy error function and back-propagation algorithm [72]. In general, first order derivative of the error function can be efficiently computed by back-propagation algorithm. However, gradient descent approach tends to converge very slow on deep neural networks. It is observed that the optimization may halt before making significant progress and the training data is under-fitted by the model [73]. In the optimization community, second order approach is usually applied for a better approximation of local curvature, such that the convergence speed can be much faster [22]. The biggest challenge of using second order approach to optimize the neural network weights is that the Hessian matrix is too huge to compute exactly and store in memory. Typically, approximations are made for computing Hessian or its inverse [1, 22]. Recently, a Hessian free approach has been proposed to optimize the deep neural networks [73], which has proved that Hessian free approach without pre-training can outperform the first order approach with pre-training. As the objective of computing Hessian is actually to revise the parameter search direction by multiplying its inverse and the gradient. If the resulting product can be effectively approximated, a lot of computing time and memory can be saved.

In order to perform Viterbi decoding, the state emission probabilities should be prepared. As DNN can only predict the state posterior probabilities, Bayes theorem can be applied:

$$p(\mathbf{o}_t|j) = \frac{p(j|\mathbf{o}_t)p(\mathbf{o}_t)}{P(j)} \propto \frac{p(j|\mathbf{o}_t)}{P(j)} \tag{2.143}$$

where $p(j|\mathbf{o}_t)$ is the context dependent state posterior, $P(j)$ is the prior probability of state j. Normalization by the prior probability $P(j)$ may not improve the recognition accuracy, but it is sometimes very important to solve the bias label problems, such as long silence segments. Therefore, it is always preferable to do so for safety. State transition probabilities of CD-GMM/HMM systems may continue being used for CD-DNN/HMM, which may be further optimized using CD-DNN/HMM to do the forward-backward alignment. At the same time, it is also possible to re-generate the training labels using CD-DNN/HMM for a better alignment between the observation and its label.

2.3.5.4 Discussion

CD-DNN/HMM has shown many great successes for various large vocabulary continuous speech recognition (LVCSR) tasks [65, 69, 74], and it has also been widely used for many commercial applications, such as Bing Voice Search, Google Voice Search. So far, the biggest disadvantage of DNN compared to GMM for LVCSR tasks is that it is much harder to make good use of large cluster machines to train them on massive data sets [69]. Besides, It also has some other limitations such that it will not always be the best solution for all applications. First, both training and decoding CD-DNN/HMM system require a modern GPU for fast matrix operation, and training CD-DNN/HMM can be significantly slower than training an equivalent CD-GMM/HMM system using the same CPU. Due to the strict hardware requirement, CD-DNN/HMM system may not be suitable for off-line application on modern mobile devices or desktop without modern GPU. As a deep nonlinear classifier, DNN uses single model to predict the posterior probabilities for all states. If acoustic patterns of few partial states change, the whole DNN system has to be retained, which is very expensive in terms of computing time. In contrast, each state in a GMM/HMM system has its own statistics, such that fast re-estimation of partial affected states becomes possible. Considering an extreme case that the acoustic space has changed, if relationship can be built between the statistics before and after the change, fast re-estimation of whole system can be achieved. To improve the adaptability of DNN, front-end feature adaptation may be applied. However, only limited improvements have been observed [75, 76]. Finally, since DNN is a discriminative model, it requires a lot of training data to determine the correct boundaries and deeply understand the acoustic data for LVCSR task. This requirement may limit its applications for those languages with limited resources [77]. In summary, CD-DNN/HMM has become a new state-of-the-art acoustic model for most large vocabulary speech recognition tasks. Improvement space still exists for both CD-DNN/HMM and CD-GMM/HMM in those complicated application contexts.

2.3.6 Cross-lingual Speech Recognition

In modern days, performance of continuous speech recognition has dramatically improved and plenty of commercial services or applications based on speech recognition technology have been used in our daily life, such as dictation system, speech translation and voice search. In typical, a state-of-the-art speaker independent large vocabulary continuous speech recognition (SI-LVCSR) system requires hundreds or thousands hours of transcribed speech data. This requirement is not a problem for those languages with great economic potentials, such as English, Chinese, French, etc. However, speech technologies are not just for making money, but also have a lot of contributions for other purposes. Considering that there exist about 7000 languages in the world, speech recognition for

some rarely studied languages may be temporarily or occasionally needed for such as humanitarian, economical or regional conflict reasons. Hence, a rapid development of an ASR system with limited resources is needed. On the other hand, some languages are extincting due to minority of speakers but may have some unknown potentials, which can be better preserved with the help speech and language technologies. For convenience, language with limited resources will be named as native (target) language, while others as foreign (source) language.

Since most languages are relatively independent of each other, which means that each language has its own grammar, syntax, phone-set, applications requiring language convention are very challenging. Therefore, an usual ASR system is language dependent to ensure that most language specific characteristics are properly learned and the best performance can be obtained. However, there are also many commons among human languages from the pronunciation aspects, which means one word in one language may be approximately pronunciated by a phone sequence from another language. For example, English word "ONE" can be pronunciated by a Chinese Character "WANG". Such similarity makes it possible to borrow the rich resources from other popular languages to build the acoustic models for a new language. Typically, multi-lingual or cross-lingual speech recognition systems are adopted. Multi-lingual speech recognition system is estimated by pooling all interesting language data and using an universal phone set. It is also called as a language independent speech recognition system, and examples includes [78, 79, 80, 81, 82]. Thus, an ASR system for a new language with limited resources can be built by simply employing a lexicon using universal phone set without any other effort. One biggest advantage is that it also works when the resource of target language is ZERO. Sometimes, language adaptation can also be applied to optimize a small amount of language specific parameters if some language specific data are available. The biggest disadvantage of multi-lingual system is that the performance may be poor since the system tries to learn an universal pattern from many different languages. In addition, the system complexity of multi-lingual system can be very high in order to model so many language specific contexts, which may lead to inefficient decoding. Alternatively, cross-lingual speech recognition is proposed, which will be discussed subsequently.

2.3.6.1 Cross-lingual Phone Mapping

One of cross-lingual speech recognition technologies is phone mapping [83, 84, 85, 86]. Mathematically speaking, the objective of cross-lingual speech recognition can be expressed as:

$$\hat{Y} = \arg\max_{Y} P(O|Y)P(Y) \tag{2.144}$$

$$= \arg\max_{Y} \left(\sum_{X} P(O|X)P(X|Y) \right) P(Y) \tag{2.145}$$

where X, Y are the phone sequences of source and target language, respectively. Instead of evaluating the likelihood of target phone sequence, $P(O|Y)$ without target acoustic models known, the objective becomes evaluating the likelihood of source phone sequence, $P(O|X)$ with well trained acoustic models. Since $P(Y)$ can be simply obtained from language models, the remaining task is to estimate $P(X|Y)$, which shows the posterior probability of source phone sequence given the target sequence, a similarity metric between two phone sequence. In practice, only the most likely source phone sequence is evaluated for fast implementation, such as

$$\hat{X} = \arg\max_{X} P(X|O) \tag{2.146}$$

$$\hat{Y} \approx \arg\max_{Y} P(\hat{X}|Y)P(Y) \tag{2.147}$$

Therefore, the decoding process is decoupled into two stages: 1). the foreign phone recognizer is used to obtain the most likely foreign phone sequence, $\hat{X}$; 2). find the most likely target phone sequence, $\hat{Y}$ given the source phone sequence $\hat{X}$. The second step is also know as Probabilistic Phone Mapping (PPM) [84]. Since only the intermediate foreign phone sequence is used to decode the final target phone sequence, many details of original observation sequence are lost. Therefore, the improvement of using PPM approaches is limited. Although temporal and spatial context expansion are applied to source phone sequence for a better representation of the observation sequence, it is difficult to build a complete context dependent phone mapping given limited data, i.e. both the source and target phone sequence are context dependent. Eventually, it will affect the system performance in large vocabulary continuous speech recognition task. Instead of using intermediate recognized foreign phone sequence, foreign phone posteriors are used to predict the native phone posteriors [85], which is later used in a hybrid system for decoding. Since only simple phone recognition task is evaluated [85], it is uncertain whether it will work well for a LVCSR task using triphone-state posteriors with limited resources.

2.3.6.2 Cross-lingual Tandem features

As mentioned in previous section, the biggest challenge is that it is difficult to learn a context dependent phone mapping with limited resources. On the other hand, feature transformation techniques do not have such issue if the target features are well discriminatively trained, such as Multiple Layer Perceptron (MLP) features [8, 87], which have been proved to be better than the traditional acoustic features in mono-lingual application. Due to the acoustic similarity among human languages, MLP trained by sufficient foreign speech data can also be able to generate discriminative features for native speech data [88, 89, 90].

Figure. 2.10 shows a typical workflow to extract cross-lingual tandem features. Foreign MLP is trained by sufficient training data and context expanded observations are used as

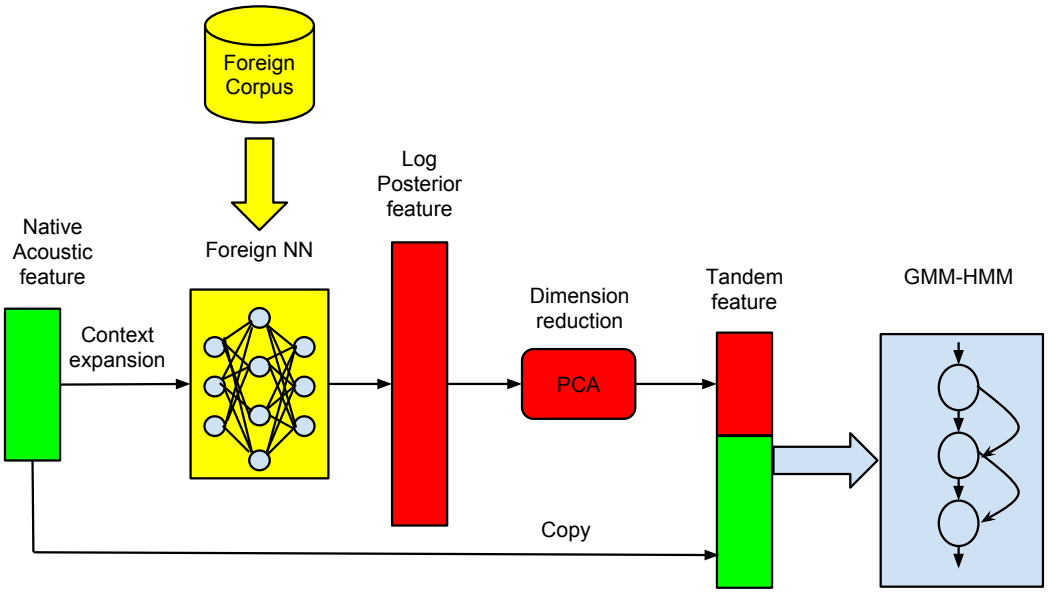

Figure 2.10: A typical workflow to extract cross-lingual tandem features

input for best performance. When foreign MLP is well trained, foreign posterior features can be predicted given the native features as input. In order to obtain a more Gaussian-like distribution [14], posterior features are transformed to be log-posteriors. After that, Principle Component Analysis (PCA) is applied to reduce the feature dimension. Finally, the conventional acoustic features and PCA projected features are concatenated to form the tandem features. However, cross-lingual tandem features do not always outperform the conventional acoustic features due to the language difference. For example, tandem features from Spanish neural networks for Chinese recognition did not perform as well as the simple baseline [90].

2.4 Summary

In this chapter, various topics related to acoustic modelling have been reviewed. First, the typical feature extraction process and its post-processing have been introduced. Then, the most widely used acoustic model, HMM together with GMM as state emission probabilities has been discussed in details. Although HMM is not a valid generative model

for speech, its effective training and decoding procedures make it become the most popular acoustic model. Nevertheless, GMM/HMM using maximum likelihood estimation has its limitations such that it usually serves as the very initial baseline system. On top of that, various state-of-the-art techniques have been proposed to improve this baseline system. Trajectory model aims to model the temporal correlation of the speech by adding history or future dependency on the state emission probabilities. Although the motivation is strongly supported, only few approaches with more computing costs have achieved significant LVCSR improvements over the conventional system. Discriminative training aims to relax the assumption that HMM is a generative model of speech. Instead of maximizing the likelihood of training data, parameters are estimated by minimizing the recognition error during discriminative training. Since discriminative training does not increase the system complexity but significantly improves the performance, it has become the standard setup of a GMM-based state-of-the-art system. So far, an assumption has been implicitly made that the training and testing conditions are similar or training conditions should contain some examples of testing condition. Considering the speech variabilities, this assumption is poorly made in practice. Adaptation and adaptive training have been introduced to solve this problem, including speaker and noise variabilities. Next, DNN/HMM has been introduced as the new state-of-the-art system. DNN can be viewed as an implicit combination of trajectory model and discriminative training, which also proves that the importance of those two topics. Finally, cross-lingual speech recognition technology has been introduced to solve the problem of building an ASR system with limited resources. In the next chapter, a new implicit trajectory model will be introduced as the main topic of this thesis.

Chapter 3

Temporally Varying Weight Regression for Speech Recognition

Standard Hidden Markov Model (HMM) assumes that successive observations are independent to one another given the state sequence. This leads to a poor trajectory model for speech. Many explicit trajectory modelling techniques have been studied in the past to improve trajectory modelling for HMM. However, these techniques do not yield promising improvements over conventional HMM systems where differential parameters and Gaussian Mixture Model have been used implicitly to circumvent the poor trajectory modelling issue of HMM. Recently, semi-parametric trajectory modelling techniques based on temporally varying model parameters such as fMPE [16] and pMPE [36] have been shown to yield promising improvements over state-of-the-art systems on large vocabulary continuous speech recognition tasks. These techniques use high dimensional posterior features derived from a long span of acoustic features to model temporally varying attributes of the speech signal. Bases corresponding to these posterior features are then discriminatively estimated to yield temporally varying mean (fMPE) and precision matrix (pMPE) parameters. Motivated by the success of fMPE and pMPE, Temporally Varying Weight Regression (TVWR), a novel semi-parametric trajectory model whose Gaussian weights varies with time, was proposed recently [11], however, its regression parameter was optimized with a weak lower bound by maximum likelihood criterion alone. In this thesis, TVWR is formulated from a formal probabilistic derivation process such that a linear regression function is naturally shown with sound constraints and necessary assumptions; parameter estimation algorithms for both ML and MPE training are derived in the TVWR framework. Part of the work has been published to the proceedings of ICASSP 2012 [11], and the remaining work has been published to IEEE/ACM transactions on Audio, Speech and Language Processing [12].

3.1 Introduction

Hidden Markov Model (HMM) [2] is widely used as acoustic models for automatic speech recognition because of its efficient training and decoding algorithm. However, such efficiency is built on a series of assumptions made by the standard HMM system. The two fundamental assumptions made by HMM are: 1) the first-order state transition probability only depends on the current state; 2) the current observation is independent of other states and observations given the current state. Considering the strong correlation between successive acoustic features within the speech utterance, these assumptions are poorly made. To circumvent the poor trajectory modelling capability of HMM, dynamic features are appended to the static acoustic features so that instantaneous trajectory information can be implicitly modelled. Furthermore, Gaussian Mixture Model (GMM) [3] has also been used to allow multiple statistics to exist within the single HMM state so that a better resolution can be achieved by dynamically switching different component. Since these two techniques can be easily implemented and lead to significant performance improvements, they already become the standard configuration of the state-of-the-art HMM-based ASR system.

However, these approaches are quite limited in terms of trajectory modelling, because statistics are still static within the state so that only a piecewise constant trajectory can be retained, which can be very far from the true trajectory of the speech signal. This motivates many research works on dynamic statistics modelling by introducing more complicated probabilistic models to relax such limitation. Some typical works are switching linear dynamical system (SLDS) [35], stochastic segment model (SSM) [91, 92], factor analyzed HMM (FAHMM) [93], polynomial segment model [24], hidden trajectory model (HTM) [94], buried Markov model (BMM) [95] and trajectory HMM [28, 29]. Some of them have been be formulated as a trajectory model with time varying parameters, as described in [36]. However, these techniques have shown little success for large vocabulary continuous speech recognition (LVCSR) [4] using maximum likelihood training. So far, discriminatively trained time varying means (fMPE [16], region dependent linear transform (RDLT) [96]) or variances (pMPE [36]), which are referred to as semi-parametric trajectory models in [36], have achieved promising improvements on LVCSR tasks. Unfortunately, the requirement of explicitly evaluating the time-varying Jacobian term due to the time-varying feature transformation in fMPE makes the likelihood function difficult to optimize. Therefore, maximum likelihood training of fMPE is not efficient [97]. In [97], a fMPE-like feature transformation is estimated using ML criterion for speaker adaptation, however, the proposed method requires multiple passes over the adaptation data and is not suitable for large scale training. On the other hand, Minimum Phone Error (MPE) [98] training of these models can be achieved, since the Jacobian term does not appear in the MPE criterion. The lack of ability to efficiently perform maximum likelihood training hinders the use of fMPE estimated models for tasks such as unsupervised speaker adaptation,

which generally do not work well with discriminative training techniques. Furthermore, both fMPE and pMPE involve quite elaborate training procedures [16, 36].

GMM is widely used to represent the HMM state emission probability density function. With sufficiently large number of Gaussian components, GMM are able to model complicated distributions. Typically, high dimensional acoustic features are used in speech recognition systems. Therefore, a large number of data samples are required to obtain a good estimation of GMM parameters. Tied-mixture or semi-continuous models can be built to achieve more compact systems. In a tied-mixture system, a pool of Gaussian components are shared by multiple HMM states. State-dependent Gaussian component weights are used to obtain distinct state distributions. Similarly, one may also consider the Gaussian components within each HMM state as the bases for the GMM distribution while the component weights vary with time. Literature GMM with input-dependent mixing weights [99] does not carry the temporal context information but aims to fit the input data locally. It is important to remember that the key of trajectory modelling is to use richer temporal context information. In our recent work [11], Temporally Varying Weight Regression (TVWR) was proposed to adjust the GMM weights according to the temporal context information, which is another example of semi-parametric trajectory models: parametric components include the static weights and regression parameters, and non-parametric components are supported by highly compact posterior features. However, previous work only implemented Maximum Likelihood estimation based on a weak lower bound and an expensive constrained optimization solver [100]. In this section, the recently proposed TVWR is re-formulated starting from a formal probabilistic derivation such that a constrained linear regression is naturally given together with necessary assumptions and sound constraints. In the parameter estimation part, the implications of TVWR formulation and assumptions are explicitly discussed such that the revised auxiliary functions of ML and MPE training for both the standard HMM parameters and new TVWR parameters can be formally derived and understood.

3.2 Temporally Varying Weight Regression

In order to avoid building a regression function for the time-varying weight empirically like fMPE, which could potentially cause Jacobian issue due to existing time-varying distribution, in this section, temporally varying weight regression (TVWR) is derived in a probabilistic fashion. Conventionally, Gaussian Mixture Model (GMM) [3] is used to model the state emission probability in HMM-based speech recognition systems. Local contextual information can be easily modelled by the conventional HMM system using a long span of acoustic features, which is named as L-HMM system for further discussion,

such that the emission probability of an HMM state j can be expressed as:

$$p(\boldsymbol{\xi}_t|j) = \sum_{m=1}^{M} \underbrace{P(m|j)}_{c_{jm}} \underbrace{p(\boldsymbol{\xi}_t|j,m)}_{\mathcal{N}(\boldsymbol{\xi}_t;\boldsymbol{\mu}_{jm}^{(\boldsymbol{\xi}_t)},\boldsymbol{\Sigma}_{jm}^{(\boldsymbol{\xi}_t)})} \tag{3.1}$$

where $\boldsymbol{\xi}_t = \{\mathbf{o}_t, \boldsymbol{\tau}_t\}$, the context variable $\boldsymbol{\tau}_t$ is given as:

$$\boldsymbol{\tau}_t = \{\mathbf{o}_{t-\delta}, \ldots, \mathbf{o}_{t-1}, \mathbf{o}_{t+1}, \ldots, \mathbf{o}_{t+\delta}\} \tag{3.2}$$

δ is the context expansion size, e.g. 4. In practice, in order to really take advantage of such long span feature for temporal correlation modelling, covariance matrix, $\boldsymbol{\Sigma}_{jm}^{(\xi)}$ should not be assumed to be diagonal any more, which however has been usually made in conventional HMM system. However, due to the high dimensionality of $\{\boldsymbol{\tau}_t, \mathbf{o}_t\}$ and limited training data, training and decoding of such GMM for a LVCSR system may become intractable. Therefore, instead of directly using above GMM to model such joint distribution, the proposed TVWR model is formulated by factorizing the joint distribution in Eq-3.1 such as:

$$p(\mathbf{o}_t, \boldsymbol{\tau}_t|j) = \sum_{m=1}^{M} \underbrace{P(m|j)p(\boldsymbol{\tau}_t|\mathbf{o}_t,j,m)}_{c_{jmt}} \underbrace{p(\mathbf{o}_t|j,m)}_{\mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_{jm},\boldsymbol{\Sigma}_{jm})} \tag{3.3}$$

This leads to a GMM distribution with stationary regular sized Gaussian components, $p(\mathbf{o}_t|j,m) = \mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_{jm},\boldsymbol{\Sigma}_{jm})$, which may keep assuming a diagonal covariance matrix for efficient training and decoding or apply other advanced precision matrix models [15, 101], but temporally varying component weights, c_{jmt}, which are obtained by scaling the time-invariant weights, c_{jm}, with the conditional probability of $\boldsymbol{\tau}_t$, i.e. $p(\boldsymbol{\tau}_t|\mathbf{o}_t,j,m)$. Directly modelling such condition dependent distribution of $\boldsymbol{\tau}_t$ can be very difficult due to its complex condition variables, i.e. a combination of discrete and continuous variables,

therefore simplification is performed by following derivation:

$$p(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m) \approx p(\boldsymbol{\tau}_t|j, m) \tag{3.4}$$

$$= \sum_{i=1}^{N} p(\boldsymbol{\tau}_t, i|j, m) \tag{3.5}$$

$$= \sum_{i=1}^{N} p(\boldsymbol{\tau}_t|i, j, m)P(i|j, m) \tag{3.6}$$

$$\approx \sum_{i=1}^{N} p(\boldsymbol{\tau}_t|i)P(i|j, m) \tag{3.7}$$

$$= \sum_{i=1}^{N} \frac{p(i|\boldsymbol{\tau}_t)p(\boldsymbol{\tau}_t)}{P(i)} P(i|j, m) \tag{3.8}$$

$$= K_t \sum_{i=1}^{N} \underbrace{\frac{p(i|\boldsymbol{\tau}_t)}{P(i)}}_{\tilde{h}_{ti}} \underbrace{P(i|j, m)}_{w_{jmi}} \tag{3.9}$$

such that $N \ll$ dimension of $\boldsymbol{\tau}_t$, where three assumptions are made:

1. $P(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m)$ is difficult to model because of the conditional dependency on the continuous high dimensional variable, $\mathbf{o}_t$. Therefore, it is assumed to be independent of the current observation given the state/component $\{j, m\}$, as in Eq-3.4. As the correlation of successive observations within the long span context variable, $\boldsymbol{\tau}_t$ can be modelled in the later stage, the detriment of missing the central observation, $\mathbf{o}_t$ for each time t should be minimal.

2. Instead of modelling the expensive distribution for each component, $p(\boldsymbol{\tau}_t|i, j, m)$, the dependency on $\{j, m\}$ is dropped so that a global model, $p(\boldsymbol{\tau}_t|i)$, can be used to represent the distribution of the contextual information, which leads to the second assumption shown in Eq-3.7. When $\boldsymbol{\tau}_t$ is partitioned via the phone/state, the dependency on $\{j, m\}$ may be carried by the latent variable. In that case, there may be no need of explicit dependency on $\{j, m\}$.

K_t is a component independent normalization term which could be ignored as it will not change the relative component likelihood. Therefore, K_t need not be explicitly computed during both training and testing. A discrete latent variable, i, is introduced to "partition" the continuous space of $\boldsymbol{\tau}_t$ into N discrete regions so that $p(\boldsymbol{\tau}_t|j, m)$ can be modelled indirectly via these discrete regions to reduce the model complexity. Specifically, the global model, $p(\boldsymbol{\tau}_t|i)$, is used to establish the probabilistic partitioning relationship between $\boldsymbol{\tau}_t$ and i. Component specific weights, $P(i|j, m)$, are then estimated to obtain the probabilistic association between i and $\{j, m\}$.

The w_{jmi} is the discrete probability of i given the component, which is also referred to as the regression parameter for the time-varying weight in the thesis. The final posterior feature $p(i|\boldsymbol{\tau}_t)$ or h_{ti} represents the probability of $\boldsymbol{\tau}_t$ belonging to the latent variable i, while $\tilde{p}(i|\boldsymbol{\tau}_t)$ or $\tilde{h}_{ti}$ is the prior normalized posterior feature. The introduction of such a latent variable and the necessary assumptions yields a compact distribution representation for $\boldsymbol{\tau}_t$. The choice of N is a trade-off between the system complexity and performance. Typically, N is set to be a small number to yield a compact model. If the latent variables are associated with the monophones or its states, then a separate probabilistic classifier, such as NN or SVM, can be trained to provide the posterior of latent variables. One advantage of using NN is that the long term temporal correlation of successive observations can be discriminatively learned in a highly nonlinear fashion, which however cannot be modelled by the conventional GMM.

Based on these assumptions, the resulting time-varying weight is given by:

$$\tilde{c}_{jmt} = P(m|j)\tilde{p}(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m) \tag{3.10}$$

where

$$\tilde{p}(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m) = K_t \sum_{i=1}^{N} \tilde{p}(i|\boldsymbol{\tau}_t) P(i|j, m) \quad \text{and} \quad \tilde{p}(i|\boldsymbol{\tau}_t) = \frac{p(i|\boldsymbol{\tau}_t)}{P(i)} \tag{3.11}$$

The following constraints are needed to ensure that the resulting joint probability, $p(\mathbf{o}_t, \boldsymbol{\tau}_t|j)$, is a valid probability distribution:

$$\sum_{m=1}^{M} c_{jm} = 1, \forall j \quad \text{and} \quad c_{jm} \geq 0, \forall j, m \tag{3.12}$$

$$\sum_{i=1}^{N} w_{jmi} = 1, \forall j, m \quad \text{and} \quad w_{jmi} \geq 0, \forall j, m, i \tag{3.13}$$

The full derivation can be found in the Appendix.A.2. Based on the simplifications in Eq-3.4,3.7, it turns out that it is not necessary to constrain the time-varying weights, $\tilde{c}_{jmt}$ to sum to one for every time frame, t. Therefore, unlike the traditional mixture component weight, $\tilde{c}_{jmt}$ does not need to have the sum-to-one property.

3.3 Parameter Estimation

In this section, parameter estimation for the time varying weights will be described. The implications of the fundamental assumptions will be explicitly discussed in deriving the auxiliary function and sufficient statistics for both Maximum Likelihood (ML) and Minimum Phone Error (MPE) criteria.

3.3.1 Maximum Likelihood Training

The maximum likelihood estimation of the TVWR parameters can be achieved by maximizing the following log likelihood function:

$$\mathcal{L}_{ML}(\mathbf{\Lambda}) = \log p(O_1^T|\Lambda) = \log \sum_{Q_1^T} p(O_1^T, Q_1^T|\Lambda) \tag{3.14}$$

where O_1^T and Q_1^T are the observation and state sequence respectively, $\mathbf{\Lambda}$ is the model parameters. After applying the *Baum-Welch* algorithm [2], the auxiliary function can be obtained as the strict lower bound of function in Eq.3.14:

$$\mathcal{Q}_{ML}(\mathbf{\Lambda}, \hat{\mathbf{\Lambda}}) = \sum_{t,j} \gamma_j^{ML}(t) \log p(\boldsymbol{\xi}_t|j) \tag{3.15}$$

where $\gamma_j^{ML}(t)$ is the state j posterior probability at time t given the current model parameter $\hat{\mathbf{\Lambda}}$ and the whole utterance.

Given the assumption in Eq-3.4, in order to optimize the component dependent parameters, the auxiliary functions becomes:

$$\begin{aligned}
\mathcal{Q}_{ML} &= \sum_{t,j,m} \gamma_{jm}^{ML}(t) \log \{c_{jm} p(\mathbf{o}_t, \boldsymbol{\tau}_t|j,m)\} && (3.16)\\
&= \sum_{t,j,m} \gamma_{jm}^{ML}(t) \log \{c_{jm} p(\boldsymbol{\tau}_t|j,m) p(\boldsymbol{\tau}_t|\mathbf{o}_t,j,m)\} && (3.17)\\
&\approx \sum_{t,j,m} \gamma_{jm}^{ML}(t) \{\log c_{jm} + \log p(\mathbf{o}_t|j,m) + \log p(\boldsymbol{\tau}_t|j,m)\} && (3.18)
\end{aligned}$$

where the revised component's ML occupancy in TVWR is:

$$\begin{aligned}
\gamma_{jm}^{ML}(t) &= \gamma_j^{ML}(t) \frac{\hat{c}_{jm} p(\boldsymbol{\xi}_t|j,m,\hat{\mathbf{\Lambda}})}{\sum_{m=1}^{M} \hat{c}_{jm} p(\boldsymbol{\xi}_t|j,m,\hat{\mathbf{\Lambda}})} \\
&= \gamma_j^{ML}(t) \frac{\tilde{c}_{jmt}(\hat{\mathbf{\Lambda}}) \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\mathbf{\Sigma}}_{jm})}{\sum_{m=1}^{M} \tilde{c}_{jmt}(\hat{\mathbf{\Lambda}}) \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\mathbf{\Sigma}}_{jm})}
\end{aligned} \tag{3.19}$$

$\hat{c}_{jm}, \tilde{c}_{jmt}(\hat{\mathbf{\Lambda}}), \hat{\boldsymbol{\mu}}_{jm}, \hat{\mathbf{\Sigma}}_{jm}$ are given by the current TVWR parameters. Hence, the standard GMM parameters can be updated using the standard formula except using the above revised component occupancy.

Given the assumption in Eq-3.7, in order to optimize the global model related parameters, the auxiliary function becomes:

$$\begin{aligned}
\mathcal{Q}_{ML} &= \sum_{t,j,m,i} \gamma_{jmi}^{ML}(t) \log \{w_{jmi} p(\boldsymbol{\tau}_t|i,j,m)\} && (3.20)\\
&\approx \sum_{t,j,m,i} \gamma_{jmi}^{ML}(t) \{\log w_{jmi} + \log p(\boldsymbol{\tau}_t|i)\} && (3.21)
\end{aligned}$$

where the new latent variable's ML occupancy[1] in TVWR is:

$$\begin{aligned}\gamma_{jmi}^{ML}(t) &= \gamma_{jm}^{ML}(t)\frac{\hat{w}_{jmi}p(\boldsymbol{\tau}_t|i,\hat{\boldsymbol{\Lambda}})}{\sum_{i=1}^{N}\hat{w}_{jmi}p(\boldsymbol{\tau}_t|i,\hat{\boldsymbol{\Lambda}})} \\ &\approx \gamma_{jm}^{ML}(t)\frac{\hat{w}_{jmi}\tilde{h}_{ti}}{\sum_{i=1}^{N}\hat{w}_{jmi}\tilde{h}_{ti}}\end{aligned} \tag{3.22}$$

In case of using neural networks to predict the posterior feature, the global model can be assumed to be trained independently. However, if GMM is applied for the global model, we can estimate the global model similarly to the standard GMM parameters. Anyway, when the global model is assumed to be trained independently for convenience, the auxiliary function to be optimized w.r.t. w_{jmi} for a particular component m is:

$$\mathcal{Q}_{ML}^{(m)} = \sum_{i=1}^{N}\beta_{jmi}^{ML}\log w_{jmi}, \quad \text{where} \quad \beta_{jmi}^{ML} = \sum_{t=1}^{T}\gamma_{jmi}^{ML}(t) \tag{3.23}$$

subject to constraint given by Eq-3.13. Lagrange function can be introduced to solve this constrained optimization problem such as:

$$\mathcal{L} = \sum_{i=1}^{N}\beta_{jmi}^{ML}\log w_{jmi} + \lambda(\sum_{i=1}^{N}w_{jmi} - 1) \tag{3.24}$$

In order to obtain the optimal solution, following equation system needs to be solved:

$$\frac{\partial\mathcal{L}}{\partial w_{jmi}} = \frac{\beta_{jmi}^{ML}}{w_{jmi}} + \lambda = 0 \tag{3.25}$$

$$\frac{\partial\mathcal{L}}{\partial\lambda} = \sum_{i}w_{jmi} - 1 = 0 \tag{3.26}$$

Finally, the update formula for the regression parameter can be obtained as:

$$w_{jmi} = \frac{\beta_{jmi}^{ML}}{\sum_{i=1}^{N}\beta_{jmi}^{ML}} \tag{3.27}$$

According to this update formula, two facts can be obtained: 1) if one uninformative posterior is shown, i.e. $p(i|\boldsymbol{\tau}_t) = P(i)$, $\forall i$, the regression value becomes component independent $\sum_{i=1}^{N}\tilde{h}_{ti}w_{jmi} = 1$; 2) if the posterior feature is time-invariant, i.e. $\tilde{p}(i|\boldsymbol{\tau}_t) = \tilde{h}_i$, $\forall i,t$ ($\tilde{h}_i = h_i/P(i)$ is not necessary to be 1), the estimated regression parameter becomes component independent, i.e. $w_{jmi} = \tilde{h}_i/\sum\tilde{h}_i, \forall j,m$. Both facts can keep the relative component likelihood of TVWR the same as the standard HMM model, which implies TVWR is able to unlearn the uninformative contextual knowledge.

[1]Note that the lower bound of the original TVWR [11] is strict at the first iteration because of the same occupancy, where w_{jmi} is initialized to be uniform, but not necessarily at the subsequent iterative training.

3.3.2 Discriminative Training

Although various discriminative training criteria have been proposed for past few years [44], in this section, only the most popular Minimum Phone Error (MPE) [98] is elaborated for estimating TVWR parameters. The MPE criterion, which measures the expected phone accuracy of all possible sentences s for each training sentence r, is given by:

$$\mathcal{F}_{MPE}(\Lambda) = \sum_{r=1}^{R} \sum_{s} \frac{p(O_r|s,\Lambda)^{\kappa} P(s)^{\kappa}}{\sum_u p(O_r|u,\Lambda)^{\kappa} P(u)^{\kappa}} A(s, s_r) \tag{3.28}$$

where O_r is sentence r specific observation sequence, $A(s, s_r)$ returns a "raw phone accuracy" of hypothesized sentence s given its reference s_r and κ is the posterior probability scaling factor. Instead of directly optimizing this objective function, which is very difficult, given assumption Eq-3.4, a weak-sense auxiliary function is derived as:

$$\begin{aligned}
\mathcal{Q}_{MPE} &= \sum_{t,j,m} \gamma_{jm}^{MPE}(t) \log \{c_{jm} p(\mathbf{o}_t, \boldsymbol{\tau}_t|j,m)\} && (3.29)\\
&= \sum_{t,j,m} \gamma_{jm}^{MPE}(t) \log \{c_{jm} p(\mathbf{o}_t|j,m) p(\boldsymbol{\tau}_t|\mathbf{o}_t,j,m)\} && (3.30)\\
&\approx \sum_{t,j,m} \gamma_{jm}^{MPE}(t) \{\log c_{jm} + \log p(\mathbf{o}_t|j,m) + \log p(\boldsymbol{\tau}_t|j,m)\} && (3.31)
\end{aligned}$$

where the revised component's MPE "occupancy" in TVWR is:

$$\gamma_{jm}^{MPE}(t) = \sum_{r=1}^{R} \sum_{q \ni t}^{Q_r} \gamma_q^{MPE} \gamma_{qjm}^{ML}(t) \tag{3.32}$$

$q \ni t$ is the phone arc containing the frame t, Q_r is the set of arcs in the training sentence r, γ_q^{MPE} is the scaled differential of the MPE objective function w.r.t. the arc log likelihood:

$$\begin{aligned}
\gamma_q^{MPE} &= \frac{1}{\kappa} \frac{\partial \mathcal{F}_{MPE}}{\partial \log p(O_r|q)} && (3.33)\\
&= \gamma_q^{ML}(c(q) - c_{avg}^r) && (3.34)
\end{aligned}$$

where γ_q^{ML} is the likelihood of the arc q as derived from a forward-backward likelihood computation over the arcs, $c(q)$ is the average accuracy $A(s : q \in s, s_r)$ of sentences passing through the arc q, and c_{avg}^r is the average $A(s, s_r)$ of all the sentences in the recognition lattice for the r-th training utterance (Note that the average is obtained by the sentence likelihood based weighted sum), and

$$\begin{aligned}
\gamma_{qjm}^{ML}(t) &= \gamma_{qj}^{ML}(t) \frac{\hat{c}_{jm} p(\boldsymbol{\xi}_t|j,m,\hat{\boldsymbol{\Lambda}})}{\sum_{m=1}^{M} \hat{c}_{jm} p(\boldsymbol{\xi}_t|j,m,\hat{\boldsymbol{\Lambda}})}\\
&\approx \gamma_{qj}^{ML}(t) \frac{\tilde{c}_{jmt}(\hat{\boldsymbol{\Lambda}}) \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{\Sigma}}_{jm})}{\sum_{m=1}^{M} \tilde{c}_{jmt}(\hat{\boldsymbol{\Lambda}}) \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{\Sigma}}_{jm})} && (3.35)
\end{aligned}$$

where $\gamma_{qj}^{ML}(t)$ is the state occupancy given the current model and arc q. Thus, the standard MPE training of the Gaussian parameters can be performed independently of w_{jmi}.

Given the assumption in Eq-3.7, in order to estimate the regression parameters of the global model, the auxiliary function to be optimized becomes:

$$\mathcal{Q}_{MPE} = \sum_{t,j,m,i} \gamma_{jmi}^{MPE}(t) \log \{w_{jmi} p(\boldsymbol{\tau}_t | i, j, m)\} \tag{3.36}$$

$$\approx \sum_{t,j,m,i} \gamma_{jmi}^{MPE}(t) \{\log w_{jmi} + \log p(\boldsymbol{\tau}_t | i)\} \tag{3.37}$$

where the new latent variable's MPE "occupancy" in TVWR is:

$$\begin{aligned} \gamma_{jmi}^{MPE}(t) &= \gamma_{jm}^{MPE}(t) \frac{\hat{w}_{jmi} p(\boldsymbol{\tau}_t | i, \hat{\boldsymbol{\Lambda}})}{\sum_{i=1}^{N} \hat{w}_{jmi} p(\boldsymbol{\tau}_t | i, \hat{\boldsymbol{\Lambda}})} \\ &= \gamma_{jm}^{MPE}(t) \frac{\hat{w}_{jmi} \tilde{h}_{ti}}{\sum_{i=1}^{N} \hat{w}_{jmi} \tilde{h}_{ti}} \end{aligned} \tag{3.38}$$

After dropping terms independent of w_{jmi}, the auxiliary function can be expressed in terms of the numerator and denominator statistics as follows:

$$\mathcal{Q}_{MPE} = \sum_{t,j,m,i} \left(\gamma_{jmi}^{num}(t) - \gamma_{jmi}^{den}(t)\right) \log w_{jmi} \tag{3.39}$$

where $\gamma_{jmi}^{num}(t)$ and $\gamma_{jmi}^{den}(t)$ are the MPE "occupancy counts" for the numerator and denominator models, respectively.

However, since optimizing the above function could cause zero weight update [44], the revised auxiliary function for a particular component m is given as follows:

$$\mathcal{Q}_{MPE}^{(m)} = \sum_{i=1}^{N} \beta_{jmi}^{num} \log w_{jmi} - \frac{\beta_{jmi}^{den}}{C} \left(\frac{w_{jmi}}{w_{jmi}^{orig}}\right)^{C} \tag{3.40}$$

where the sufficient statistics are given by:

$$\beta_{jmi}^{num} = \sum_{t=1}^{T} \gamma_{jmi}^{num}(t) \quad \text{and} \quad \beta_{jmi}^{den} = \sum_{t=1}^{T} \gamma_{jmi}^{den}(t) \tag{3.41}$$

w_{jmi}^{orig} is the current parameter, and $C > 0$ is a smoothing constant for controlling the learning speed. After this, the same algorithm for updating GMM weights proposed in [44] can be employed to solve the above constrained optimization problem. Particularly, if $C = 1$, Lagrange multiplier can be applied for a faster implementation (see Appendix.A.3).

3.3.3 I-Smoothing

Typically, the regression parameters are first estimated using maximum likelihood criterion before performing discriminative training. In some cases, due to insufficient training data, I-Smoothing [98] is used to ensure robust discriminative estimation of the parameters. The same I-Smoothing technique used for the standard weight update can also be extended to the regression parameters of TVWR. The auxiliary function given in Eq-3.40 for discriminative training of w_{jmi} can be interpolated with the ML auxiliary function such as

$$\hat{\mathrm{Q}}^{(m)}_{MPE} = \sum_{i=1}^{N} \beta^{num}_{jmi} \log w_{jmi} - \frac{\beta^{den}_{jmi}}{C} \left(\frac{w_{jmi}}{w^{orig}_{jmi}} \right)^{C} + \tau^{R} \sum_{i=1}^{N} \frac{\beta^{ML}_{jmi}}{\sum_{i=1}^{N} \beta^{ML}_{jmi}} \log w_{jmi} \tag{3.42}$$

$$= \sum_{i=1}^{N} \hat{\beta}^{num}_{jmi} \log w_{jmi} - \frac{\beta^{den}_{jmi}}{C} \left(\frac{w_{jmi}}{w^{orig}_{jmi}} \right)^{C} \tag{3.43}$$

where the revised occupancy is given as

$$\hat{\beta}^{num}_{jmi} = \beta^{num}_{jmi} + \tau^{R} \frac{\beta^{ML}_{jmi}}{\sum_{i=1}^{N} \beta^{ML}_{jmi}} \tag{3.44}$$

τ^{R} is the smoothing constant to control the learning rate of w_{jmi}. In general, I-Smoothing can improve the generalization of the parameters, which may lead to a better performance.

3.4 Comparison to fMPE

It is interesting to note that fMPE [16] can also be recast as a similar time-varying weight structure given in Eq-3.3. Given the formulation of fMPE:

$$\mathbf{y}_t = \mathbf{o}_t + \mathbf{M}\mathbf{h}_t \tag{3.45}$$

where $\mathbf{h}_t$ is the high dimensional posterior feature vector [1], whose dimension can be hundreds of thousands, and $\mathbf{M}$ is the projection matrix to learn. The state output probability in fMPE is presented as:

$$p(\mathbf{y}_t|j) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}^{(\mathbf{y})}_{jm}, \boldsymbol{\Sigma}^{(\mathbf{y})}_{jm})$$

$$= \sum_{m=1}^{M} c_{jmt} \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}^{(\mathbf{y})}_{jm}, \boldsymbol{\Sigma}^{(\mathbf{y})}_{jm}) \tag{3.46}$$

[1] Note that $\mathbf{h}_t$ is dependent on both $\mathbf{o}_t$ and $\boldsymbol{\tau}_t$ in fMPE. However, in TVWR, $\mathbf{h}_t$ is derived from $\boldsymbol{\tau}_t$ only.

where

$$c_{jmt} = c_{jm} \exp\left\{\mathbf{h}_t^T \mathbf{M}^T \boldsymbol{\Sigma}_{jm}^{(y)-1} \left(\boldsymbol{\mu}_{jm}^{(y)} - \mathbf{o}_t - \frac{1}{2}\mathbf{M}\mathbf{h}_t\right)\right\} \tag{3.47}$$

Since it is impractical to estimate such high dimensional projection matrix using Newton's method, a steepest gradient descent method is applied such as:

$$\mathbf{M}_{ij} = \mathbf{M}_{ij} + \alpha_{ij} \frac{\partial F_{MPE}}{\partial \mathbf{M}_{ij}} \tag{3.48}$$

where $\mathbf{M}_{ij}$ is the row-i and column-j element of matrix $\mathbf{M}$, α_{ij} is the learning rate. Typically, the learning rate of gradient method has to be defined to satisfy the Wolfe Conditions [22], however, it is very expensive to evaluate the objective function for each learning rate, which requires a go-through of the whole training data. In practice, fMPE [16] uses an approximated approach by separating the gradient based on its sign for each frame such as:

$$\alpha_{ij} = \frac{\sigma_i}{E(p_{ij} + n_{ij})} \tag{3.49}$$

where σ_i is the standard deviation of i-th feature dimension, E is the constant to control the learning rate for the matrix, which has to be manually adjusted by several trials, positive and negative gradients are accumulated as part of sufficient statistics during training, respectively:

$$p_{ij} = \sum_{t=1}^{T} \max\left(\frac{\partial F_{MPE}}{\partial \mathbf{M}_{ij}}, 0\right) \tag{3.50}$$

$$n_{ij} = \sum_{t=1}^{T} \max\left(-\frac{\partial F_{MPE}}{\partial \mathbf{M}_{ij}}, 0\right) \tag{3.51}$$

Although these two state output probabilities given in Eq-3.3 and Eq-3.46 have similar parametric structure, the physical meanings of the fundamental parameter are very different:

1. In the TVWR formulation, time-varying attributes are incorporated as an independent factor such that the Gaussian components are unaffected by the time-varying formulation. On the other hand, fMPE formulation leads to a time-varying Jacobian in the likelihood function. Therefore, TVWR parameters can be easily estimated using both the ML and MPE criteria. However, fMPE parameters cannot be efficiently estimated using the ML objective function because sufficient statistics cannot be accumulated due to the time-varying Jacobian term. The work in [97] attempts to apply ML estimation to fMPE-like model to perform speaker adaptation, which can be tractable as the amount of adaptation data is usually small. This is not an issue with fMPE itself because the time-varying Jacobian terms cancel out in the MPE objective function.

2. The Gaussian means/variances in these two methods are estimated differently: fMPE is based on the nonlinear transformed features while TVWR is based on the original features. There are several advantages in keeping the Gaussian means/variances in the original feature space. A) The Gaussian means/variances and its weights are independent such that the changes in the regression parameters (and hence the time-varying weights) do not affect the Gaussian means and variances. Therefore, there is no need to go through 3 passes over the data to accumulate the sufficient statistics like fMPE. B) It is possible to apply model-based noise compensation techniques, such as VTS [58] and PMC [57]. This is not possible for fMPE since it is hard to derive representations for the noise impact on the speech parameterized by non-linear transformed features [102]. Therefore, only feature enhancement technique, such as Stereo-based Stochastic Mapping (SSM) [103], can be applied to improve the noise robustness of fMPE. However, stereo training data is not always available in practice.

Although some uninformative posterior, i.e. the one with $h_{ti} = P(i)$, could produce zero gradient at those time-frames during training because of the indirect differential of fMPE, the parameter update formula of fMPE is based on the overall gradient such that zero gradient information is buried during decoding. In an extreme case, if all the posterior features during decoding are uninformative, which are not during training, TVWR can still perform as good as the standard system while fMPE will become totally messed up. This inconsistency between training and decoding in fMPE would lead to much more degradation than TVWR if the deployment environment is different from the training environment.

3.5 Experimental Results

In this section, the experimental results are conducted for speaker independent Large Vocabulary Continuous Speech Recognition (LVCSR) on Wall Street Journal corpus (SI284 WSJ0+WSJ1), which contains about 81 hours of training speech data (37k utterances and 284 speakers). The 20k open vocabulary recognition task (NIST Nov'92 WSJ0) with 333 utterances is used as test set for evaluation. All the acoustic models used in this work are based on decision tree state-clustered triphone systems with 5769 distinct states. Each triphone unit is modelled by a 3-state left-to-right HMM. These acoustic models are trained on 39 dimensional MFCC features, including 12 static coefficients, energy and the first two differentials. All the reported recognition results are based on a bigram full decoding followed by a trigram lattice-rescoring using HTK [5].

The posterior feature, $p(i|\boldsymbol{\tau}_t)$ used in TVWR is synthesized by a 3-layer feedforward neural network [1]: the hidden layer consists of 1000 neurons with sigmoid activation func-

[1]Using ICSI quicknet software package, http://www.icsi.berkeley.edu/speech/qn.htm

tion; the input to the neural network is a sequence of MFCC features spanning a window of 8 frames. The neural network was trained using frame-based cross-entropy criterion to predict 120 monophone states with softmax activated output units. The first 30k utterances of this corpus are used for training and the rest are used for cross validation. In this report, monophone-state posterior features will be used to regress the time-varying weight for TVWR system. The posterior feature used in fMPE is 5769 tied-state posterior, which is obtained for each frame from a well estimated HMM system with one mixture per state. Gaussian clustering and minimum posterior threshold, i.e. 0.01 in this report, were used for efficient calculation. Acoustic context expansion based on the approach presented in [16, 36] was applied to provide richer temporal information. Eventually, the final posterior feature used in fMPE is with context expansion window (± 3) and spanning 19 frames window. Preliminary results showed that higher dimension of posterior feature caused over training issue in this specific task.

TVWR system is initialized with $w_{jmi} = \frac{1}{N}$ if it starts from a standard HMM system, which is mathematically equivalent to the corresponding HMM system with static weights only. In this experiment, the prior probability was also assumed as $P(i) = \frac{1}{N}$ for convenience. In case of MPE training of TVWR systems, it is possible to obtain the initial model using either an ML-trained HMM or ML-trained TVWR system.

3.5.1 ML Training of TVWR

In case of ML training, ML-trained HMM systems with various number of components were obtained as the baseline by 8 iteration estimations. After that, ML-trained TVWR systems were obtained by another 4 iteration estimations with all the model parameters updated. As a reference, L-HMM systems using the long span features, i.e. 39×9 dimensional features, were estimated based on the alignment of 7th iteration of respective HMM systems for efficient training; TVWR0 systems with only regression parameters updated were also estimated for a better comparison. As the time varying normalization term K_t is not explicitly evaluated, the true likelihood during ML training of TVWR is not presenting. Therefore, only the recognition results shown in Table. 3.1 are to be discussed.

L-HMM systems with dramatically more parameters are without over-training issue and performs slightly better than the HMM baseline system, which tells that many Gaussian parameters of L-HMM are redundant if only with diagonal covariance matrix. When compared to the HMM baseline system, TVWR shows consistently significant [1] improvements but various amount for different system complexity. Since GMM is one example of implicit trajectory models, which can offer a better resolution by switching different components within the state, the improvement by TVWR decreased as expected. However, GMM is a considerably weak trajectory model as it is estimated based on the

[1] Statistical significance were computed at $p \leq 0.05$ using NIST SCTK Scoring Tookit.

HMM assumptions. Explicitly modelling the dependency of successive observations using neural networks gives more accurate information for the speech recognition. On the other hand, TVWR0 achieved consistent improvement over the baseline system but is marginally inferior to TVWR. These tell that the Gaussian parameters in TVWR are indeed the statistics for describing the distribution of the original features and TVWR is able to improve the forward-backward alignment accuracy of the standard HMM system, respectively. When a very large number of components are applied, i.e. 64 mixtures per state, both HMM and TVWR systems become suffering from the over-training issue. However, the fact that TVWR is able to improve the over-trained HMM system tells that re-adjusting GMM weights is not just to build a non-stationary distribution but also to correct the probability from a over-trained model.

#Gaussian per state	WER(%)			
	HMM	L-HMM	TVWR0	TVWR
2	14.83	14.12	12.58	12.09
4	12.33	12.16	11.32	11.06
8	10.70	10.88	10.07	9.96
16	10.38	10.30	9.71	9.60
32	9.82	9.73	9.39	9.39
64	10.67	9.76	9.75	9.73

Table 3.1: *Comparison of 20k task performance for ML trained HMM and TVWR systems.*

3.5.2 MPE Training of TVWR

All the experiments of discriminative training under MPE criterion are conducted based on systems with 16 Gaussian components per state [1] I-Smoothing using ML as the prior is applied to all the Gaussian parameters and regression parameters. The smoothing constants are chosen to improve training stability. The training lattices were generated using a unigram language model trained by a large number of text files from the corpus, excluding the acoustic training sentences. Before moving on to discuss the performance of MPE trained TVWR, several related systems are described in Table. 3.2. The initial models for these system are given. For systems with MPE training, τ^I, τ^W and τ^R are also given for I-smoothing Gaussian parameters, GMM static weights and TVWR regression parameters, respectively.

[1]Note that MPE baseline system with 32 mixtures per state has difficulty to achieve significant gain over the ML-trained HMM system due to the insufficient training data for discriminative training.

System	Starting point	I-Smoothing
MPE	HMM.ML	$\tau^I = 100, \tau^W = 10$
fMPE	HMM.ML	-
fMPE+MPE	fMPE	$\tau^I = 100, \tau^W = 10$
TVWR.MPE0	TVWR.ML	$\tau^W = 10, \tau^R = 0$
TVWR.MPE1	HMM.ML	$\tau^I = 100, \tau^W = 10, \tau^R = 10$
TVWR.MPE2	TVWR.ML	$\tau^I = 100, \tau^W = 10, \tau^R = 0$

Table 3.2: *Different discriminatively trained system configuration descriptions.*

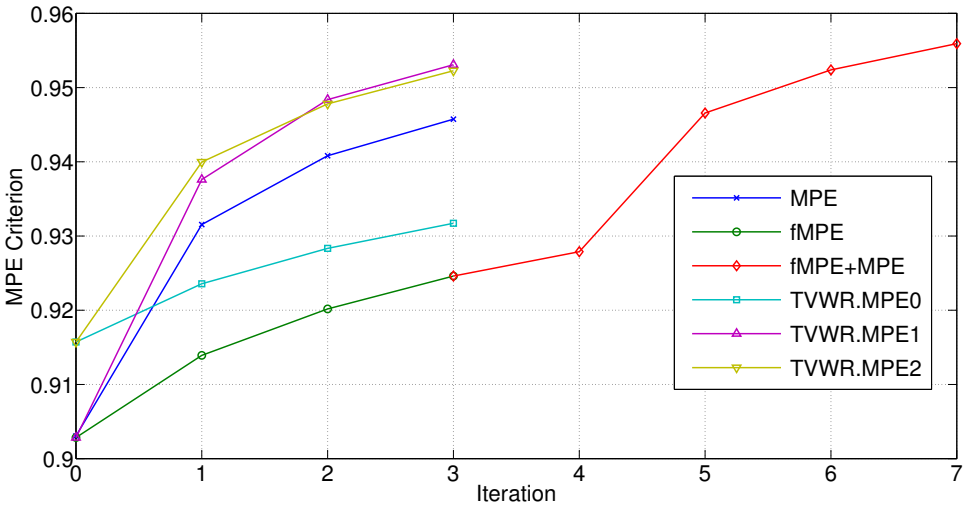

Figure 3.1: *Comparison of MPE criterion for each discriminatively trained systems.*

Next, the improvements of MPE criteria with training iterations for various systems are plotted in Figure. 3.1. There are several important information that can be observed from this figure:

- Discriminative training achieved consistent improvements in terms of MPE criterion with increasing training iterations for all the systems
- Previously, it was shown that TVWR outperformed the ML-trained HMM system for ML training. Therefore, systems using TVWR.ML as the starting point have a better initial MPE objective function value.
- After discriminative training of all parameters, all systems eventually outperformed the baseline MPE system in terms of the MPE criterion.
- For fMPE and TVWR.MPE0, the standard Gaussian parameters were not estimated using MPE. The final MPE scores for these systems were significantly lower

than the other systems. Nevertheless, TVWR.MPE0 achieved consistently higher MPE scores compared to fMPE. Note that the comparison of these two systems indicates the ability to model the time varying attributes: fMPE models them using time varying mean while TVWR.MPE0 models the time varying Gaussian weights. Note that the ability to estimate TVWR parameters using the ML criterion gives TVWR.MPE0 a better head start.

- Usually, better MPE score is a good sign of better test performance, so this figure already shows some potential capabilities of discriminatively trained TVWR.

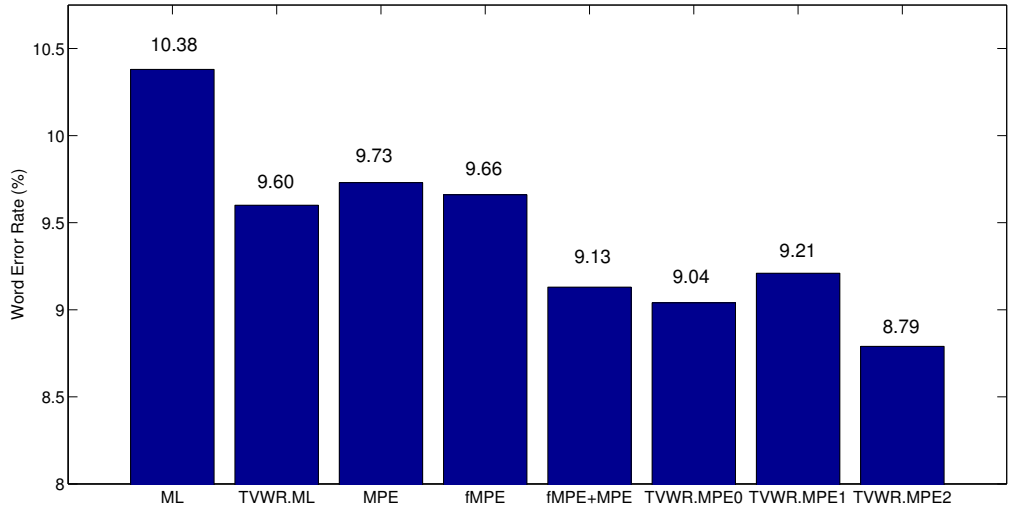

Figure 3.2: *Comparison of 20k task for various discriminatively trained systems.*

The recognition results of 20k tasks for different discriminatively trained systems are reported in Figure. 3.2. When Gaussian parameters are estimated by maximum likelihood, fMPE shows comparable performance to the MPE-trained HMM system and significantly better than the ML-trained HMM system, which tells that trajectory modelling is an important factor for a more accurate acoustic model. Similar to fMPE, TVWR.MPE0 with ML trained Gaussian parameters, however, shows 0.6% absolute further improvement over fMPE. These results show that the discriminative training algorithm of TVWR is well implemented and indeed able to obtain a promising improvement over the ML trained TVWR system. It is also interesting to note that TVWR.MPE0 can show a comparable performance to fMPE+MPE, which shows discriminatively estimated TVWR regression parameters can be very powerful. On the other hand, the gain of discriminative training TVWR parameters without updating Gaussian parameters could benefit noise robust speech recognition if model compensation approaches are applied.

After all the model parameters are discriminatively estimated, TVWR.MPE2 shows 0.9% absolute improvement over the MPE-trained HMM system; it is also marginally better performance than fMPE+MPE, which would be expected as both of them used the similar posterior features to model the same non-stationary attributes of the speech. On the other hand, considering the much simpler formulation and training procedures, TVWR can a better alternative of fMPE for temporal correlation modelling in practice. When comparing different starting points, TVWR.MPE2 shows much better performance than TVWR.MPE1, which suggests the discriminative training of TVWR following its ML training could significantly improve the robustness and performance of the final model without I-Smoothing. Lastly, the behavior of discriminative updating Gaussian parameters is slightly different for fMPE and TVWR: TVWR.MPE2 only slightly better than TVWR.MPE0, while fMPE+MPE is much better than fMPE. Such difference tells that most gains of TVWR are from the regression parameter, while fMPE heavily depends on the Gaussian parameters.

3.5.3 I-Smoothing for TVWR

In order to find out the effect of I-Smoothing for TVWR parameters, TVWR.MPE1 systems with the same $\tau^I = 100$ but various τ^R are iteratively evaluated based on the 20k test task, as shown in Figure. 3.3. When no smoothing is applied ($\tau^R = 0$), the system achieves the lowest WER at the first iteration, but fluctuates iteratively and gives worst results in the end. After applying I-smoothing, the learning rate is reduced but a more consistent WER reduction can be achieved. Eventually, after 4 iterations, both $\tau^R = 10$ and $\tau^R = 20$ lead to a much better performance. These results show that I-Smoothing is a very important factor for discriminative training of TVWR parameters to improve its generalization, particularly in case of starting from a standard HMM system.

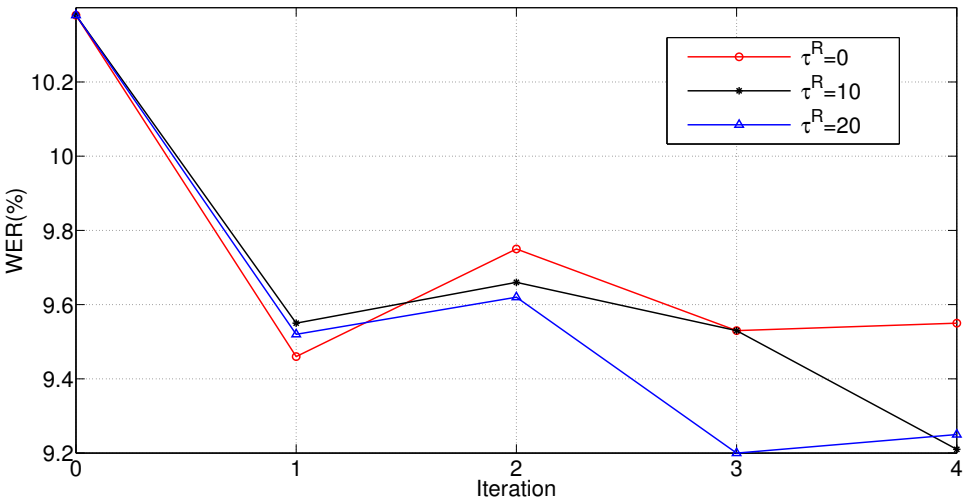

Figure 3.3: *Iterative evaluation of TVWR.MPE1 with different I-Smoothing constant* τ^R.

3.5.4 Noisy Speech Recognition

In order to investigate the feasibility of TVWR on more challenging data, noisy speech recognition on Aurora 4 [104] is evaluated. In this task, only the multi-condition training data with 14 hours of speech data with 16k sampling rate are used, including 6 noises artificially corrupted data at 15 dB average SNR and various microphones. MFCC features contains 12 static cepstral coefficients, zero-th coefficient and first two derivatives. For a better performance, cepstral mean and variance normalization (CMVN) is applied for feature enhancement. The baseline is a decision tree state-clustered triphone system with 3226 states and 16 mixtures per state. The fMPE baseline is estimated based on the tied-state posterior feature with a ± 3 context expansion window. The same monophone-state posterior feature predicted from neural networks with the same configuration as the first experiment is used for TVWR system. Evaluation data includes 14 test sets with 330 utterances per set for 5k closed vocabulary recognition task (NIST Nov'92 WSJ0), which are partitioned into four sets: set A is clean, set B is corrupted by 6 types of noise, set C has channel distortion, and set D has both channel distortion and 6 additive noises. The average SNR in the noise corrupted test sets is 10 dB. The results in Table. 3.3 are obtained using a bigram full decoding followed by a trigram rescoring. Note that all TVWR systems will update both HMM parameters and regression parameters. I-smoothing constants used here are $\tau^I = 50$, $\tau^W = 10$, $\tau^R = 10$.

System	A	B	C	D	Average
ML	8.41	13.07	14.83	26.62	18.67
TVWR.ML	6.65	11.70	13.71	26.09	17.65
MPE	7.10	11.79	13.45	26.01	17.67
fMPE+MPE (S1)	6.15	11.19	12.48	25.32	16.98
TVWR.MPE (S2)	5.75	10.88	12.55	25.12	16.74
CNC (S1+S2)	5.94	10.49	11.92	24.08	16.09

Table 3.3: *Aurora 4 recognition results for various multi-condition trained systems.*

Our ML-trained HMM baseline achieved an average WER performance of 18.67% using trigram (23.03% using bigram). The bigram result is comparable to the 21.4% performance reported in [105], which used the ETSI advanced front-end (AFE) [106]. When ML training is performed, TVWR.ML shows promising improvements on the test set A, B and C over the baseline system, but relatively small gain for the speech with both channel distortion and additive noise. Since half of the training data is without channel distortion and the other half with channel distortion from many different microphones, TVWR system may have difficulties learning the speech pattern from all microphones due to the limited training data. After MPE training, MPE-trained HMM system gave similar

performance compared to TVWR.ML. Furthermore, both fMPE+MPE and TVWR.MPE outperforms the MPE system. Again, we observe larger gains on set A, B and C. Overall, TVWR gave a consistent absolute WER reduction of about 1% using both ML and MPE training.

Despite the similar performance of fMPE+MPE (S1) and TVWR.MPE (S2), the Confusion Network Combination (CNC) [107] of the two systems gave a further absolute WER reduction of 0.65%. However, CNC does not yield further improvement on set A (clean), which is possibly because fMPE and TVWR learned some similar time-varying information due to clear pattern of clean speech. These results imply that these two systems are considerably different in modelling more noisy speech data.

3.6 Summary

This chapter has introduced an implicit trajectory model using Temporally Varying Weight Regression (TVWR) to learn the importance of Gaussian components under different acoustic contexts. This allows the temporally varying attributes of speech to be better recognized. Instead of modelling the temporal correlations directly using a long span of acoustic features, contextual information is implicitly incorporated into TVWR using posterior features. As a result, TVWR is able to model non-stationary GMM distributions whose temporally varying Gaussian component weights are obtained through regression with the posterior features. Although TVWR is similar to fMPE in terms of modeling time-varying model parameters using posteriors derived from a long span of acoustic features, the underlying formulation and model parameterization are different. Specifically, TVWR follows a proper probabilistic formulation that yields a much simpler parameter estimation compared to fMPE. Moreover, both maximum likelihood and discriminative training parameter estimation formulae can be derived. Experiments were conducted on the Wall Street Journal (CSR-WSJ0+WSJ1) corpora for 20k open vocabulary continuous speech recognition and the Aurora 4 corpus for 5k closed vocabulary noisy speech recognition. Experimental results show that TVWR achieves better performance compared to the standard HMM system for both maximum likelihood and minimum phone error training. Moreover, the discriminatively trained TVWR models also achieved comparable (or marginally better) performance compared to fMPE.

Chapter 4

Multi-stream TVWR for Cross-lingual Speech Recognition

Building a robust Automatic Speech Recognition (ASR) system with limited resources is a very challenging task due to many speech variabilities in practice. Multilingual and cross-lingual speech recognition techniques are commonly used for this task. This section investigates Temporally Varying Weight Regression (TVWR) method for cross-lingual speech recognition. TVWR uses posterior features to implicitly model the long-term temporal structures in acoustic patterns. By leveraging on the well-trained foreign recognizers, high quality monophone/state posteriors can be easily incorporated into TVWR to boost the ASR performance on low-resource languages. Furthermore, multi-stream TVWR is proposed, where multiple sets of posterior features are used to incorporate richer (temporal and spatial) context information. Finally, a separate state-tying for the TVWR regression parameters is used to better utilize the more reliable posterior features. Experimental results are evaluated for English and Malay speech recognition with limited resources. By using the Czech, Hungarian and Russian posterior features, TVWR was found to consistently outperform the tandem systems trained on the same features. This work has been published to the proceedings of ASRU workshop 2013 [13].

4.1 Introduction

Recently, multilingual and cross-lingual speech recognition has attracted many researchers due to its challenges and practical applications. This task is particularly designed to build an Automatic Speech Recognition (ASR) system with limited resources, particularly limited transcribed speech data. Although the number of tied triphone states can be reduced to provide sufficient training data for each physical state, the performance could be dramatically decreased due to the poor modelling of acoustic contexts. When the accuracy of the ASR system is low, it becomes difficult to utilize massive un-transcribed speech

data. In order to improve the performance of an ASR system with limited resources, researchers began to investigate borrowing the rich resources from other languages due to the similar acoustic characteristics among human languages. For convenience, language with limited resources will be named as native (target) language, while others as foreign (source) language.

One popular approach is to train a multilingual ASR system [78, 80, 81, 82] by pooling resources from all related languages. The ASR system for a target language can be easily obtained by defining a new lexicon using the universal phone set. For a better performance, language adaptation is also applied by optimizing small number of language specific parameters. However, the complexity of the resulting multilingual system may be very high in order to model all the different contexts and language specific patterns, which can lead to inefficient decoding. Other researchers have interests in finding a probabilistic phone mapping [83, 84, 85, 86, 108] between the source language and target language. Thus, it may be applied to adapt the acoustic models before decoding [85, 86], or translate the foreign phone sequence after decoding [83, 84, 108]. The biggest challenge of phone mapping is that it is difficult to robustly map context dependent phone sets given very limited resources. Lastly, tandem features [88, 89, 90] based on well-trained foreign-language neural networks phone recognizers have shown promising results for cross-lingual speech recognition. However, not all tandem features from foreign language can outperform the native acoustic features, e.g. tandem features from Spanish neural networks for Chinese recognition does not perform as good as baseline [90].

Temporally Varying Weight Regression [11] was recently proposed to improve the temporal correlation modelling for Hidden Markov Models (HMM). It extends the conventional HMM by incorporating posterior features trained on long-span acoustic features to model temporally-varying GMM weights. In this chapter, TVWR is applied to cross-lingual speech recognition by leveraging on well-trained foreign monophone/state recognizers to produce high quality posterior features. In addition, multi-stream TVWR is proposed where multiple sets of posterior features are used to incorporate richer spatial and temporal context information. Finally, a separate tree-based state-tying is applied to the TVWR regression parameters to better exploit the more reliable foreign posterior features.

4.2 Multi-stream TVWR

In the conventional TVWR setup, monophone/state is introduced to represent the latent variable i such that the system complexity can keep relatively low. Since foreign monophone/state posterior features are used, the performance improvement might be limited by the language differences. In order to further improve the performance under limited-resource condition, multi-stream TVWR is proposed to introduce richer context

information to regress the time-varying weights. Specifically, i is now a context-rich latent variable. Without losing generalization, i is defined as a structured variable:

$$i = \{i_1, i_2, \dots i_c \dots i_C\} \tag{4.1}$$

$$\mathcal{N} = \mathcal{N}_1 \times \mathcal{N}_2 \cdots \mathcal{N}_c \cdots \mathcal{N}_C \tag{4.2}$$

where i is now composed of C context-specific "sub-variables" and $\mathcal{N}_c$ is the set of the c-th sub-variable, i_c. This can be viewed as employing multiple partition strategies such that a much higher resolution of the acoustic space can be obtained for better discrimination. However, even with $C = 2$ or 3, the resulting set $\mathcal{N}$ can be very large. Therefore, it is difficult to estimate the joint posterior probability and there will be a lot of regression parameters to estimate. To circumvent this problem, the context-specific latent variables are assumed to be independent such that the joint posterior probabilities, $P(i_1, \dots, i_C|\tau_t)$, and the corresponding regression parameters, $P(i_1, \dots, i_C|j, m)$, can be factorized as follows:

$$p(i_1 \dots i_C|\boldsymbol{\tau}_t) \approx p(i_1|\boldsymbol{\tau}_t) \cdots p(i_C|\boldsymbol{\tau}_t) \tag{4.3}$$

$$p(i_1 \dots i_C|j, m) \approx p(i_1|j, m) \cdots p(i_C|j, m) \tag{4.4}$$

This assumption significantly reduces the system complexity and makes the TVWR formulation tractable. This leads to a multi-stream TVWR where Eq.3.9 can be rewritten as:

$$\tilde{p}(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m) \approx K_t \prod_{c=1}^{C} \sum_{i_c \in \mathcal{N}_c} \tilde{p}(i_c|\boldsymbol{\tau}_t) p(i_c|j, m) \quad \text{and} \quad \tilde{p}(i_c|\boldsymbol{\tau}_t) = \frac{p(i|\boldsymbol{\tau}_t)}{P(i_c)} \tag{4.5}$$

This formulation can be illustrated by a system diagram in Figure.4.1, where the time-varying weight is obtained by a product of multiple regressions. Note that multi-stream TVWR is different from multi-stream HMM system. In multi-stream HMM system, each state has multiple stream acoustic features and each stream is represented by a GMM, while every state in multi-stream TVWR has only one stream acoustic feature represented by a single GMM. In the following subsections, multi-stream TVWR will be used to incorporate both temporal and spatial contexts.

4.2.1 Temporal Context Expansion

In continuous speech, the sound of a phone can be easily influenced by its preceding and succeeding phones, a phenomenon called *co-articulation*, this correlation can be modelled by introducing a cross word triphone/state. However, this important information is lost when using monophone/state to represent the latent variable i in TVWR. Therefore, a temporal context dependent latent variable i is introduced by setting $C = 3$ such that i_1, i_2, i_3 can be used to indicate left, middle and right monophone/state of current frame, respectively. Since these "sub-variables" are from the same monophone/state set but

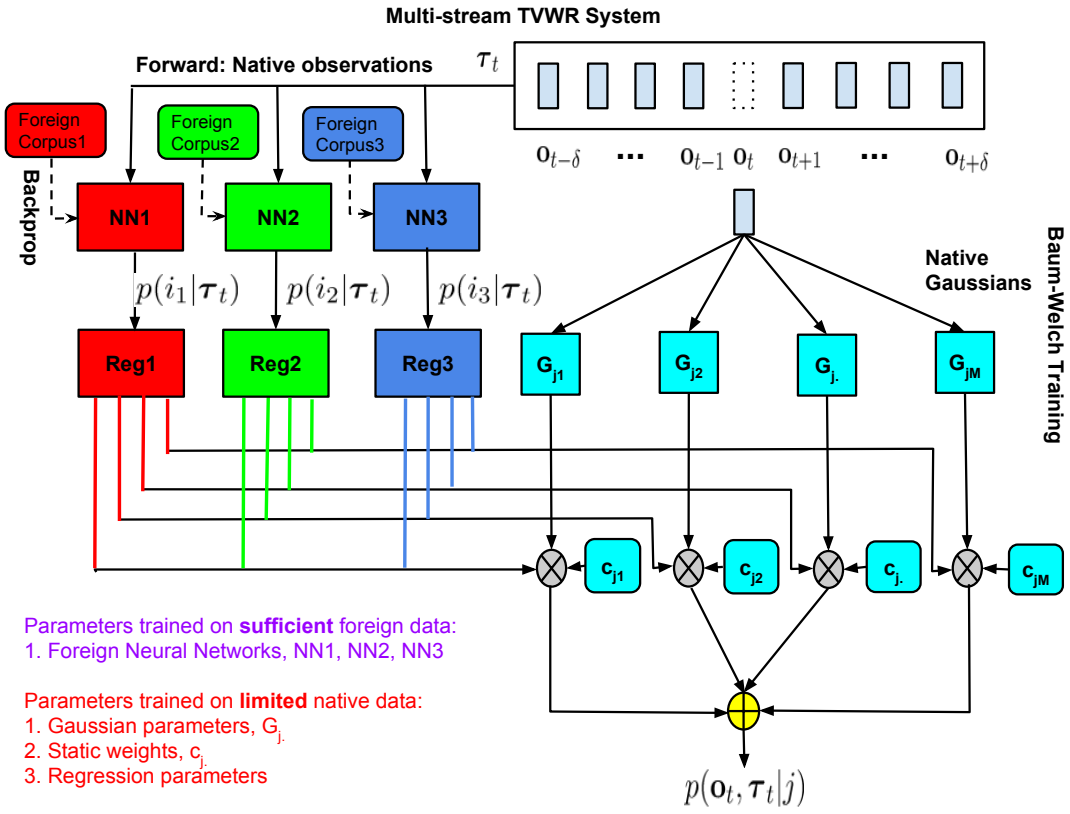

Figure 4.1: *A system diagram of multi-stream TVWR for cross lingual speech recognition.*

with different position information, X_1, X_2, X_3 are literally the same without considering context position.

Instead of using three separate recognizers to produce three sets of posteriors, only one recognizer is used to predict the middle monophone/state posterior probabilities. The corresponding left and right posterior probabilities are derived from the sequence of middle monophone/state posterior probabilities as follows: starting from the current frame, search left and right until the identity of the monophone/state with the largest posterior probability changes and use the corresponding posterior probabilities as the left and right posteriors. As a results, the left posterior feature, $p(i_1|\boldsymbol{\tau}_t)$ is given by $p(i_2|\boldsymbol{\tau}_{t-\phi})$ where:

$$\arg\max_{i_2} p(i_2|\boldsymbol{\tau}_{t-\phi}) \neq \arg\max_{i_2} p(i_2|\boldsymbol{\tau}_t) \tag{4.6}$$

$$\arg\max_{i_2} p(i_2|\boldsymbol{\tau}_{t-k}) = \arg\max_{i_2} p(i_2|\boldsymbol{\tau}_t) \quad k \in [1, \phi) \tag{4.7}$$

At the same time, the right posterior feature, $p(x_3|\boldsymbol{\tau}_t)$ can also be obtained in a similar way. Since silence does not need context, its left and right posteriors are assumed to be the same as the middle posteriors.

4.2.2 Spatial Context Expansion

Alternatively, multiple monophone/state predictors from different foreign languages can be applied to build a spatial context. In general, multiple foreign languages with more differences can lead to a better discrimination, since they are more likely to be complementary for each other. Spatial context can be more useful when each individual foreign language does not provide a good prediction of monophone/state posterior features. Therefore, C in Eq.4.5 will represent the total number of foreign languages to be applied, while $\mathcal{N}_c$ is the monophone/state set for c-th language.

4.2.3 Parameter Estimation

After ignoring independent terms, the auxiliary function w.r.t. regression parameters can be written as:

$$\begin{aligned} Q(\Lambda, \hat{\boldsymbol{\Lambda}}) &= \sum_{t,j,m} \gamma_{jm}(t) \log \Big(\prod_{c=1}^{C} \sum_{i_c \in \mathcal{N}_c} p(\boldsymbol{\tau}_t | i_c) P(i_c | j, m) \Big) \\ &\geq \sum_{t,j,m,c,i_c} \gamma_{jmi_c}(t) \Big(\log P(i_c | j, m) + \log p(\boldsymbol{\tau}_t | i_c) \Big) \end{aligned} \tag{4.8}$$

where the component occupancy is now given as:

$$\gamma_{jm}(t) = \gamma_j(t) \frac{\tilde{c}_{jmt}(\hat{\boldsymbol{\Lambda}}) p(\mathbf{o}_t | j, m)}{\sum_{m=1}^{M} \tilde{c}_{jmt}(\hat{\boldsymbol{\Lambda}}) p(\mathbf{o}_t | j, m)} \tag{4.9}$$

$\gamma_j(t)$ is the state occupancy at time t given the current model $\hat{\boldsymbol{\Lambda}}$, $\hat{c}_{jm}(t)$ is the current time-varying weight of multi-stream TVWR and

$$\gamma_{jmi_c}(t) = \gamma_{jm}(t) \frac{P(i_c | j, m, \hat{\boldsymbol{\Lambda}}) p(\boldsymbol{\tau}_t | i_c)}{\sum_{i_c \in \mathcal{N}_c} p(i_c | j, m, \hat{\boldsymbol{\Lambda}}) p(\boldsymbol{\tau}_t | i_c)} \tag{4.10}$$

$$\approx \gamma_{jm}(t) \frac{P(i_c | j, m, \hat{\boldsymbol{\Lambda}}) \tilde{p}(i_c | \boldsymbol{\tau}_t)}{\sum_{i_c \in \mathcal{N}_c} P(i_c | j, m, \hat{\boldsymbol{\Lambda}}) \tilde{p}(i_c | \boldsymbol{\tau}_t)} \tag{4.11}$$

The optimal estimation can be then obtained by using Lagrange multiplier such that:

$$P(i_c | j, m) = \frac{\sum_t \gamma_{jmi_c}(t)}{\sum_{i_c \in \mathcal{N}_c} \sum_t \gamma_{jmi_c}(t)} \quad \forall c \in C, \quad i_c \in \mathcal{N}_c \tag{4.12}$$

Note that this update formula is similar to applying the standard TVWR estimation (by setting $C = 1$) to each stream independently, except that the component occupancy is calculated using the multi-stream TVWR system when performing forward-backward calculations in the E-step of the Baum-Welch training.

4.3 State Clustering for Regression Parameters

Complexity of a context dependent HMM system is usually controlled by using the decision tree state clustering technique [109]. When training a system with limited resources, the number of distinct triphone state clusters is kept small to ensure robust estimation of all the parameters associated with the tied states. However, this may limit the potential of the TVWR system where the regression parameters cannot take full advantage of the high quality posterior features. To alleviate this problem, a separate tree-based clustering algorithm is applied to the TVWR regression parameters so that the model complexity with respect to the regression parameters can be controlled independent of the regular GMM parameters.

4.3.1 Tree-based State Clustering

Decision Tree-based state clustering [109] has been widely used to control the complexity of a triphone system due to the limited training data and massive triphone states/components. The number of clustered states needs to be carefully chosen to balance the phone disambiguation and model robustness due to limited data. As posterior features have different characteristics to conventional acoustic features, the number of clustered states or state clustering strategy for them can be very different. In order to emphasize such difference, a separate state clustering approach is proposed to cluster the states for regression parameters. With an additional decision tree for state clustering, better disambiguation for phones may be achieved as illustrated in Figure.4.2.

Due to the limited training data, state clustering algorithm will be performed on the system with only one mixture per state. The essence of the tree-based state clustering algorithm is the derivation of the likelihood increase as a result of splitting a state cluster into two. This allows the appropriate questions to be chosen for each node when constructing the decision tree. The following state-clustering derivation for the TVWR regression parameters is largely based on the framework given in [110]. The auxiliary function to be maximized with respect to the TVWR regression parameters can be written as:

$$Q(\Lambda, \hat{\Lambda}) = \sum_t \sum_{j \in \mathcal{J}} \gamma_j(t) \log \Big(\sum_{i \in \mathcal{N}} p(\boldsymbol{\tau}_t | i) P(i|j) \Big) \tag{4.13}$$

$$\geq \sum_t \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{N}} \gamma_{ji}(t) \Big(\log P(i|j) + \log p(\boldsymbol{\tau}_t | i) \Big) \tag{4.14}$$

$$= \sum_t \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{N}} \gamma_{ji}(t) \log P(i|j) + K(\mathcal{J}) \tag{4.15}$$

where $\gamma_j(t)$ is the state occupancy at time t given the current model $\hat{\Lambda}$ and transcription, $\mathcal{J}$ is a state cluster including all triphone states, $\gamma_{ji}(t)$ can be similarly obtained by setting

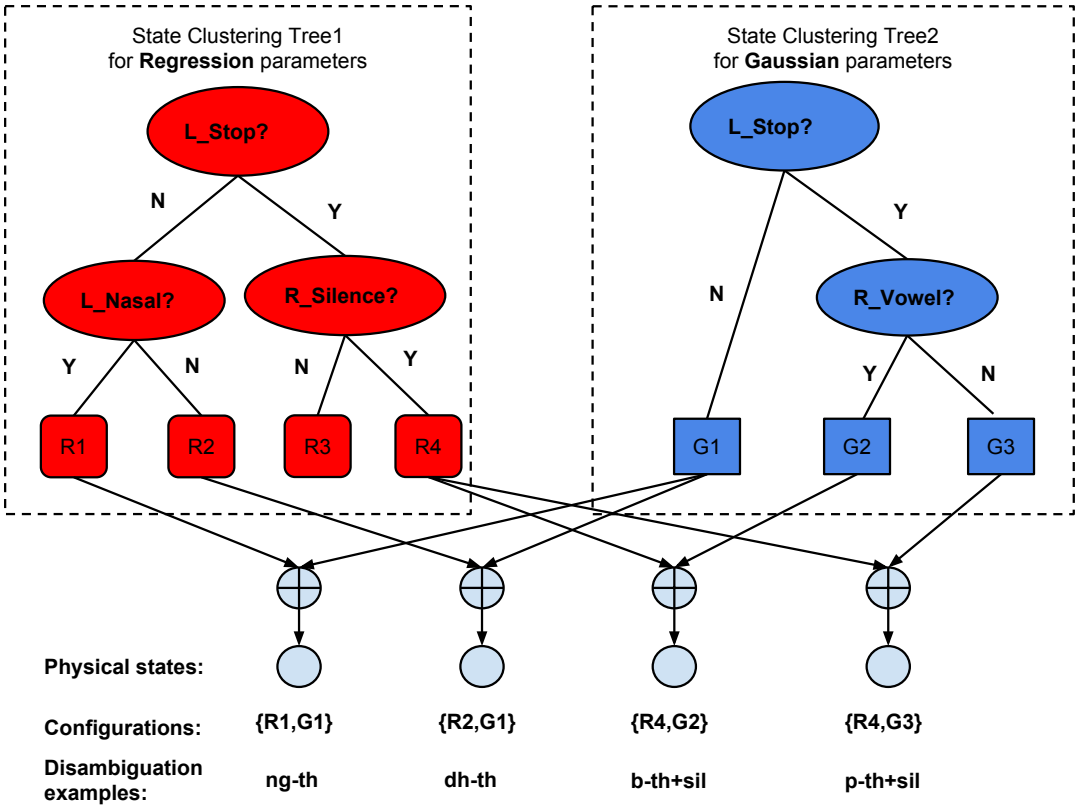

Figure 4.2: *A demonstration of disambiguating different phones with an additional decision tree.*

$m = 1$ in Eq-3.22, and

$$K(\mathcal{J}) = \sum_t \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{N}} \gamma_{ji}(t) \log p(\boldsymbol{\tau}_t | i) \tag{4.16}$$

the optimal solution of $P(i|j)$ can be similarly found by setting $m = 1$ in Eq-3.27. Assuming that the alignment, $\gamma_{ji}(t)$ is unchanged during state clustering, the auxiliary likelihood function for a state cluster $\mathcal{S}$ can be obtained as:

$$Q(\mathcal{S}) = \sum_{i \in \mathcal{N}} \sum_{j \in S} \beta_j P(i|j) \log P(i|\mathcal{S}) + K(\mathcal{S}) \tag{4.17}$$

where $\beta_j = \sum_{t,i} \gamma_{ji}(t)$ is the state occupancy, and the regression parameter for cluster $\mathcal{S}$ is given as

$$P(i|\mathcal{S}) = \frac{\sum_{j \in \mathcal{S}} \beta_j P(i|j)}{\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{S}} \beta_j P(i|j)} \tag{4.18}$$

the question is selected to maximize the following function:

$$\Delta Q_q = Q(\mathcal{S}_y(q)) + Q(\mathcal{S}_n(q)) - Q(\mathcal{S}) \tag{4.19}$$

where $\mathcal{S}$ is the initial state cluster, $\mathcal{S}_y(q), \mathcal{S}_n(q)$ are the split state clusters for "yes" and "no" answers, respectively. Given the fact that

$$K(\mathcal{S}_y(q)) + K(\mathcal{S}_n(q)) - K(\mathcal{S}) = 0 \tag{4.20}$$

the objective function, ΔQ_q will only depend on the regression parameters of each cluster, $P(i|\mathcal{S})$, $P(i|\mathcal{S}_y(q))$ and $P(i|\mathcal{S}_n(q))$.

4.3.2 Implementation Details

After introducing a second state clustering strategy, the physical states are now defined by two independent sets of parameters: one is associated to the Gaussian parameters and another is to the regression parameters. Therefore, two physical states may share the same Gaussian parameters, which however will be recognized as a single physical state for caching the state emission probabilities in HTK toolkit. During training and decoding of the conventional TVWR, if one logical state is found to be accessed before, the cached state emission probability will be used directly. After introducing a second state clustering, we have to check whether the state emission probability has been calculated considering both static Gaussian emission probabilities and time-varying GMM weights. Therefore, the caching strategy in the HTK toolkit has to be modified according to the new definition of physical state after introducing a second state clustering for the regression parameters. However, for convenience, we can simply change the state-level cache to the component-level cache such that there is no need to verify whether two logical states are connected to the same physical state.

4.4 Experimental Results

The experiments are conducted for two native (target) language recognition tasks: 1) 5k close vocabulary English speech recognition, 2) 22k open vocabulary Malay speech recognition. The full English dataset (WSJ0) contains 7k+ utterances (15 hours) with 84 speakers for training, and 330 utterances with 8 speakers for testing, while the full Malay dataset contains 35k+ utterances (74.5 hours) with 28 speakers for training, and 600 utterances with 6 speakers for testing. Both English and Malay corpora are reading speech recorded in clean environments. 39-dimension MFCC features are used for both corpora, including 13 static parameters and first two derivatives as dynamic parameters. A 3-state left-to-right HMM is applied as the acoustic model for each triphone, and tree based state clustering is applied to cluster triphone states. To perform the recognition, both use full bigram decoding and trigram lattice rescoring, while four-gram lattice rescoring is additionally employed for Malay recognition.

For cross-lingual experiments, an 1.2 hours of English subset are extracted, including 500 utterances with 5 speakers. 6 hours of Malay subset are extracted, including 3k utterances with 6 speakers. Three foreign (source) language resources are employed, including Czech (CZ), Hungarian (HU) and Russian (RU). Specifically, three foreign phone recognizers [111] well trained by respective telephone speech data are employed. For clarification, English, Malay, CZ, HU, RU have 40, 33, 45, 61, 52 monophones, respectively. In order to use these three foreign phone recognizers, all speech waveform files were down-sampled to 8kHz, which are also used to extract acoustic features for consistency. In our experiments, phone recognizers were used to generate respective monophone/state posterior features instead of monophone/state sequence.

4.4.1 Baseline Mono-lingual Recognition

English HMM fullset baseline system is obtained with 3151 tied states and 16 mixtures per state; Malay HMM fullset baseline is estimated with 5043 tied states and 16 mixture per state. Due to limited data, English HMM subset baseline contains only 445 tied states and 8 mixtures per state, while Malay HMM subset baseline contains 1178 tied states and 4 mixtures per state, which will be the default number of components for all subsequent experiments if not mentioned explicitly. Performance degradation was observed by further increasing the number of mixtures using Maximum Likelihood criterion. In order to build a TVWR subset baseline, two neural networks are estimated to predict English and Malay monophone posterior features. Both neural networks are obtained by training a 3-layer neural network using the subset and quicknet [1] with $\delta = 4$ and 1000 hidden units. TVWR subset baseline is estimated by starting from respective HMM subset baseline and using respective monophone posterior features. As shown in Table. 4.1, dramatical performance degradation was observed on both HMM and TVWR subset baseline systems. Although TVWR obtained consistent improvements over respective HMM subset baseline, its performance is still far from the HMM fullset baseline. These results clearly show that the performance of both HMM and TVWR is sensitive to the amount of available training data.

Target	HMM_full	HMM_sub	TVWR_sub
English	**6.9**	24.3	22.1
Malay	**13.1**	24.4	23.1

Table 4.1: *WER(%) performance of HMM and TVWR fullset/subset baseline systems for English and Malay speech recognition.*

[1] ICSI quicknet software package, http://www.icsi.berkeley.edu/speech/qn.htm

4.4.2 Tandem Cross-lingual Recognition

To obtain tandem features, three foreign phone recognizers were used to generate respective monophone-state posterior features for the English and Malay subset. Log posterior features are then obtained for a more Gaussian-like distribution [14]. Principle Component Analysis (PCA) was applied to obtain 13-dimension features, which was concatenated to the original 39-dimension MFCC features. Tandem systems using 52-dimension features were estimated using two-model re-estimation and 4 iteration ML estimation. The best performance on English subset was found on the tandem systems with 8 mixtures per state, while Malay subset was with 12 mixtures per state. As shown in Table. 4.2, different tandem system performs slightly differently but generally obtained significantly improvements over the HMM baseline. For English speech recognition, tandem systems using single foreign phone recognizer achieved 7-9% absolute improvements, while Russian language with best performance probably has more commons to the target English language. However, for Malay speech recognition, the absolute improvements is only 4% by using single foreign recognizer, while Hungarian seems more similar to the target Malay language. After combining three tandem systems, further 2-3% absolute improvements are observed for both English and Malay languages. These results show that tandem features using foreign language phone recognizers can help improve the performance of these two target languages with limited resources.

Target	CZ	HU	RU	CZ⊗HU⊗RU
English	16.7	16.3	15.4	**14.1**
Malay	20.4	19.9	20.1	**17.2**

Table 4.2: *WER(%) performance of various tandem systems with limited resources for target English and Malay speech recognition.*

4.4.3 TVWR Cross-lingual Recognition

As shown in Table. 4.3, all TVWR systems using foreign posteriors outperformed both HMM and TVWR subset baselines in Table. 4.1. When no context expansion is performed, 6-7% absolute improvements for English speech recognition over HMM subset baseline are observed using single stream of posterior features, while 3-4% are observed for Malay speech recognition. This tells that using a well trained foreign phone recognizer can provide a better partition for the acoustic space for TVWR. However, when compared to tandem systems in Table. 4.2, TVWR without context expansion is consistently inferior to tandem systems. This may be because TVWR depends more on unreliable GMM by MFCC features.

Target: English	w/o temporal	w/ temporal
CZ	17.7	13.0
HU	17.8	11.9
RU	17.9	13.4
spatial context	12.1	**9.8**

Target: Malay	w/o temporal	w/ temporal
CZ	21.9	18.1
HU	20.8	17.7
RU	18.0	16.7
spatial context	16.2	**14.5**

Table 4.3: *WER(%) performance of TVWR systems with or without context expansion for target English and Malay speech recognition.*

After applying temporal/spatial context expansion, multi-stream TVWR consistently outperformed both conventional TVWR systems and tandem systems. When compared to TVWR without context expansion, 4-6% absolute improvements over respective TVWR using single foreign posteriors are observed for English language, while 3-4% absolute improvements are observed for Malay language. This shows that temporal context expansion can significantly improve TVWR system performance without suffering over-fitting issue despite introducing many regression parameters. When compared to the individual tandem systems in Table. 4.2, 2-4% absolute improvements are shown for English, while 2-3% absolute improvements for Malay. Particularly, TVWR using single HU for English and single RU for Malay already shows better than multiple stream tandem systems. These results show that multi-stream TVWR with temporal context expansion can learn more information from single stream of posterior features. TVWR with spatial context expansion by three languages performs similar to the best temporal context expansion based TVWR, i.e. HU for English and RU for Malay, which shows spatial context expansion may have a more robust acoustic partition, while temporal context expansion is more sensitive to the difference between source and target languages. After combining both temporal and spatial context, another 1-2% absolute improvements are observed, which tells that temporal and spatial context are different and complementary.

Last, multi-stream TVWR with a second state clustering method is evaluated. In order to obtain strong Gaussian bases for TVWR system, the number of tied states for GMM parameters is reduced to about 330 for English (8 mix per state) and 900 for Malay (4 mix per state), while the number of tied states for TVWR parameters increased to about 1.2-1.3k for English and 3.0-3.4k for Malay. When combining both temporal and spatial context, 1.9k and 4.8k tied states for TVWR parameters are used for English (8 mix per

state) and Malay (8 mix per state), respectively. Recognition results for various TVWR systems are reported in Table. 4.4. First, after introducing a second state clustering method, consistent improvements are found for all TVWR systems. However, the amount of improvements varies as foreign language. Generally speaking, foreign posteriors with better performance in Table. 4.3 can gain more by introducing more tied states. Since the rational of introducing more tied states for TVWR parameters is that posteriors are more reliable than acoustic features, this method may not work well if foreign posteriors are not reliable enough. Finally, combination of temporal and spatial context with more tied states achieved very close performance to the HMM fullset baseline, i.e. 1-2% difference. However, it is important to note that discriminative training is applied to obtain posterior features for TVWR, which can definitely lead to a better HMM baseline.

Target	w/ temporal			spatial	temporal +spatial
	CZ	HU	RU		
English	11.7	11.3	12.6	10.6	**8.7**
Malay	18.0	16.9	15.6	14.9	**13.9**

Table 4.4: *WER(%) performance of various multi-stream TVWR systems with a second state clustering and limited resources for target English and Malay speech recognition.*

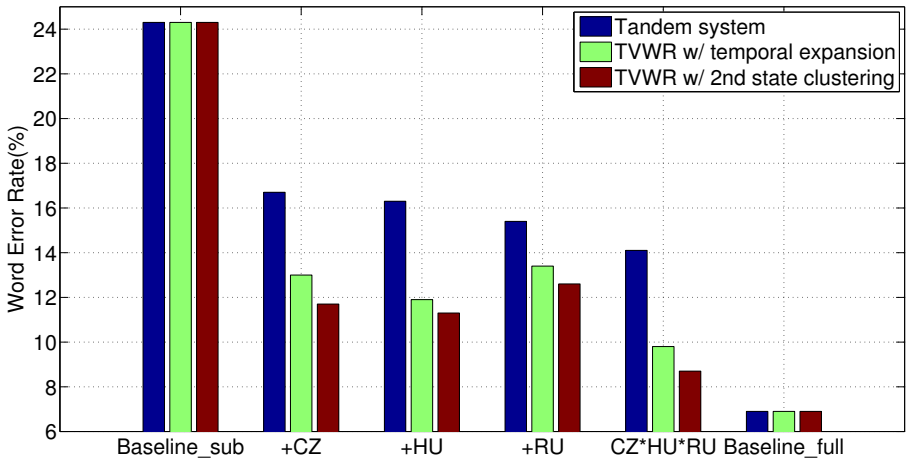

Figure 4.3: *A summarized performance comparison of various systems using 1h English training data.*

In summary, various systems are compared in Figure. 4.3 and Figure. 4.4 for English and Malay speech recognition, respectively. As can be seen, TVWR has shown consistent improvements over the tandem systems for both English and Malay speech recognition. It is worth noting that the gain of incorporating different foreign resources are different.

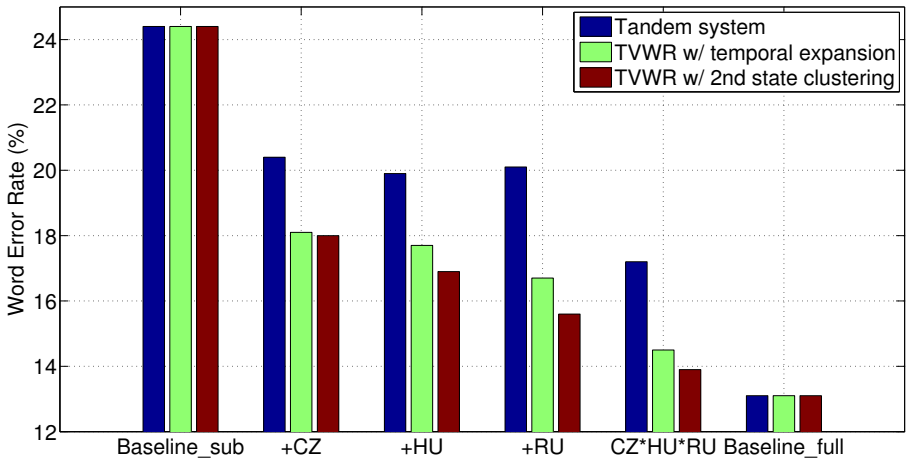

Figure 4.4: *A summarized performance comparison of various systems using 6h Malay training data.*

It tells that the common attributes of the native and foreign languages varies. For example, Hungarian is closer to English while Russian is closer to Malay. If multiple foreign resources are available, it is interesting to investigate and only use some of the most similar/relevant foreign languages. However, if the system complexity is not a concern, it might be preferable to incorporate all available resources for the best performance. Consistent and significant improvements show that TVWR is an effective approach for cross-lingual speech recognition.

4.5 Summary

In this section, proposed TVWR is investigated for cross-lingual speech recognition under limited resources. First, various foreign monophone/state posterior features are employed to replace the native unreliable features so that a better acoustic partition can be obtained. Second, multi-stream TVWR is proposed by incorporating much richer temporal and spatial context information for a better representation of the context variable. Third, a separate state clustering algorithm for the TVWR regression parameters is proposed to introduce more distinct triphone state with reliable regression parameters. Various TVWR systems were evaluated for English and Malay speech recognition with limited resources. TVWR systems using foreign monophone/state posterior features have shown significant improvements over both HMM and tandem systems. Introducing multi-stream TVWR and more tied states can obtain further improvement, which results in less than 2% inferior to respective English and Malay HMM fullset baselines.

Chapter 5

TVWR: An approach to Combine the GMM and the DNN

Recently, context-dependent Deep Neural Network (CD-DNN) has been found to significantly outperform Gaussian Mixture Model (GMM) for various large vocabulary continuous speech recognition tasks. DNN parameter estimation is primarily based on discriminative criteria, which is much more sensitive to label errors than maximum likelihood and therefore may be less reliable for unsupervised adaptation. Although various DNN adaptation techniques have been proposed, they still have limitations. On the other hand, many robust adaptation techniques that have been developed and proven to work well for GMM/HMM systems cannot be easily applied to DNNs. This chapter proposes a novel method of combining DNN and GMM using the Temporally Varying Weight Regression framework to take advantage of the superior performance of the DNNs and the robust adaptability of the GMMs. This section addresses the issue of incorporating the high-dimensional CD-DNN posteriors into this framework without dramatically increasing the system complexity. Experimental results on a broadcast news large vocabulary transcription task show that the proposed GMM+DNN/HMM system achieved significant performance gain over the baseline DNN/HMM system. With additional unsupervised speaker adaptation, the best GMM+DNN/HMM system obtained about 20% relative improvements over the DNN/HMM baseline. This work has been accepted to be published to the proceeding of ICASSP 2014 [112].

5.1 Introduction

For decades, GMM has been used as the representation of the HMM state emission probability due to its efficient training and decoding algorithms [113].In the conventional

GMM/HMM system, the state emission probability is modelled as:

$$p(\mathbf{o}_t|j) = \sum_{m=1}^{M} c_{jm} p(\mathbf{o}_t|j,m) \tag{5.1}$$

where j is the HMM state, $\mathbf{o}_t$ is the observation at time t, M is the number of Gaussian components per state, c_{jm} is the static component weight and $p(\mathbf{o}_t|j,m)$ is a Gaussian distribution. Due to the observation independence assumption and the use of diagonal covariance matrices for the Gaussian components (for better efficiency), the inter-frame and intra-frame correlations are poorly modelled by the GMM/HMM systems, which limits its performance to some extent. Nevertheless, effective adaptation techniques, such as the MLLR [48], have been developed to achieve reliable unsupervised speaker adaptation.

DNN is a general purpose machine learning model that is capable of learning the complex nonlinear function to map a long span of acoustic features into high quality CD state posterior probabilities. The state emission probability in a DNN/HMM hybrid system is given as:

$$p(\boldsymbol{\xi}_t|j) \propto \frac{p(j|\boldsymbol{\xi}_t)}{P(j)}, \quad \boldsymbol{\xi}_t = \{\mathbf{o}_{t-\delta} \ldots \mathbf{o}_t \ldots \mathbf{o}_{t+\delta}\} \tag{5.2}$$

where $P(j)$ and $p(j|\boldsymbol{\xi}_t)$ are the prior and posterior probability of state j respectively and $\boldsymbol{\xi}_t$ is the long span acoustic features. Context-dependent Deep Neural Network (CD-DNN) [64] has been reported to outperform various conventional Gaussian Mixture Models (GMM) [113] based Hidden Markov Model (HMM) [9] systems by a large margin for many large vocabulary continuous speech recognition (LVCSR) tasks [65, 69]. DNN uses a long span of acoustic features as input so that both rich inter-frame and intra-frame information can be modelled for better discrimination. The multiple layers of nonlinear transformation allows the complex relationship between the acoustic features and the context-dependent HMM states to be effectively learned. However, unlike the GMM approach where each triphone state is represented by a GMM, a single DNN is used to simultaneously predict the posterior probabilities of *all* the states. It is difficult to interpret the DNN parameters in a meaningful manner. There is no clear and effective way of adapting the DNN parameters. Moreover, the DNN parameters are typically estimated discriminatively using the cross-entropy criterion which is more sensitive to label errors. Instead of trying to interpret the acoustic meaning of DNN parameters, speaker code [114, 115] may be discriminatively learned as an additional input for part of [114] or all [115] DNN layers. Hence, the DNN weights corresponding to the speaker code can provide additional speaker information for the succeeding DNN layer. However, as the speaker code estimation is discriminative, supervised adaptation data is required, which may not robust for unsupervised adaptation. By contrast, many advanced adaptation techniques, such as Maximum Likelihood Linear Regression (MLLR) [48] and Maximum A

Posteriori (MAP) [113], have been developed and shown to work well for the GMM/HMM systems. In particular, these methods are based on the generative training paradigm, which is more robust for unsupervised adaptation.

Many workarounds have been reported to take advantage of both the GMMs and DNNs in the literatures. Some researchers suggested using the DNN to extract better discriminative features, such as the tandem features [14, 17, 116, 117]. In order to develop feasible tandem features for GMM training, the high dimensional CD-DNN posteriors have to be projected to a lower dimension, which inevitably causes information loss. Consistent performance degradation of the tandem systems using ML training has been observed in various reports [17, 116]. Further, discriminative training and unsupervised speaker adaptation have been successfully applied to the tandem systems, which achieved superior performance compared to the hybrid SI DNN/HMM system [116]. Others have also proposed to use the adapted acoustic features based on the fMLLR transformation obtained from the GMM system [76]. However, adaptive training of DNN/HMM is expected to avoid inconsistent acoustic features between training and testing, which may not be feasible if speaker information is not available during training. More recently, DNN adaptation using I-vector [118] was also proposed. I-vector [119] has been widely used for speaker verification or recognition. As it represents a speakers identity, it has been proved to be useful to provide it as an additional input for DNN. However, as I-vector is a relatively low dimensional representation of speaker, it is not scalable to the amount of adaptation data during decoding. In addition, I-vector alone is just comparable to simple fMLLR adaptation for DNN.

In this section, GMM+DNN/HMM is proposed as a novel system that combines the GMM and DNN using the Temporally Varying Weight Regression (TVWR) framework [11, 12]. Based on this framework, a regression model is trained to transform the DNN posteriors into the time-varying scaling factors for the Gaussian weights. However, directly incorporating the high-dimensional CD-DNN posterior features will lead to a substantial increase in the number of regression parameters. This section will present some solutions to address this issue.

5.2 Combining GMM and DNN

As previously mentioned, the DNNs are able to predict high quality discriminative CD posteriors while the GMMs can be reliably adapted in an unsupervised manner using MLLR. Therefore, to get the best of both worlds, this section proposes combining the GMMs and DNNs using the TVWR framework [11, 12]. According to this framework,

the state output probability of the long span acoustic features is given as:

$$p(\boldsymbol{\xi}_t|j) \propto \sum_{m=1}^{M} c_{jm} \sum_{i=1}^{N} \frac{p(i|\boldsymbol{\tau}_t)}{P(i)} P(i|j,m) p(\mathbf{o}_t|j,m) \tag{5.3}$$

$$\propto \sum_{m=1}^{M} c_{jm} \sum_{i=1}^{N} \tilde{p}(i|\boldsymbol{\tau}_t) P(i|j,m) p(\mathbf{o}_t|j,m) \tag{5.4}$$

where $\boldsymbol{\tau}_t = \{\mathbf{o}_{t-\delta} \dots \mathbf{o}_{t-1}, \mathbf{o}_{t+1} \dots \mathbf{o}_{t+\delta}\}$ denotes the contexts of the current observation, i is the latent variable to partition the acoustic space, $p(i|\boldsymbol{\tau}_t)$ is the posterior feature, $P(i|j,m)$ is the regression parameter. M and N correspond to the number of Gaussian components and the number of latent variables, respectively. As the latent variable may be associated to high dimensional context dependent phone states, its prior is explicitly presented to be consistent with DNN decoding. For convenience, normalized posterior feature is defined as: $\tilde{p}(i|\boldsymbol{\tau}_t) = p(i|\boldsymbol{\tau}_t)/P(i)$. Hence, the auxiliary function of Maximum Likelihood for updating regression parameters is revised as:

$$\mathcal{Q}(\boldsymbol{\Lambda}, \hat{\boldsymbol{\Lambda}}) = \sum_{t,j,m,i} \gamma_{jmi}(t) \log w_{jmi} \tag{5.5}$$

where the latent variable occupancy is re-written as:

$$\gamma_{jmi}(t) = \gamma_{jm}(t) \frac{\hat{w}_{jmi} p(\boldsymbol{\tau}_t|i, \hat{\boldsymbol{\Lambda}})}{\sum_{i=1}^{N} \hat{w}_{jmi} p(\boldsymbol{\tau}_t|i, \hat{\boldsymbol{\Lambda}})}$$

$$\approx \gamma_{jm}(t) \frac{\hat{w}_{jmi} \tilde{p}(i|\boldsymbol{\tau}_t)}{\sum_{i=1}^{N} \hat{w}_{jmi} \tilde{p}(i|\boldsymbol{\tau}_t)} \tag{5.6}$$

and the component occupancy is revised as:

$$\gamma_{jm}(t) = \gamma_j(t) \frac{\tilde{c}_{jmt}(\hat{\boldsymbol{\Lambda}}) \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{\Sigma}}_{jm})}{\sum_{m=1}^{M} \tilde{c}_{jmt}(\hat{\boldsymbol{\Lambda}}) \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{\Sigma}}_{jm})} \tag{5.7}$$

$$= \gamma_j(t) \frac{\left(\sum_{i=1}^{N} \hat{w}_{jmi} \tilde{p}(i|\boldsymbol{\tau}_t)\right) \hat{c}_{jm} \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{\Sigma}}_{jm})}{\sum_{m=1}^{M} \left(\sum_{i=1}^{N} \hat{w}_{jmi} \tilde{p}(i|\boldsymbol{\tau}_t)\right) \hat{c}_{jm} \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_{jm}, \hat{\boldsymbol{\Sigma}}_{jm})} \tag{5.8}$$

Similarly modification for those occupancies in discriminative training needs to be made in case of using normalized posterior features. As shown in Figure. 5.1, the long span acoustic feature is decomposed into two parts. Firstly, the regular-sized observation $\mathbf{o}_t$ is modelled by the conventional Gaussian components, where MLLR adaptation can be easily applied. Secondly, the latent variable i is associated with the clustered CD states so that $P(i|\boldsymbol{\tau}_t)$ can be predicted using a DNN. However, directly incorporating the high-dimensional CD-DNN posteriors (in the order of thousands) leads to a large number of regression parameters, $P(i|j,m)$. This will result in expensive computation for both training and decoding and may cause over-fitting. In the next section, two solutions will be presented to address this issue without compromising the model efficiency.

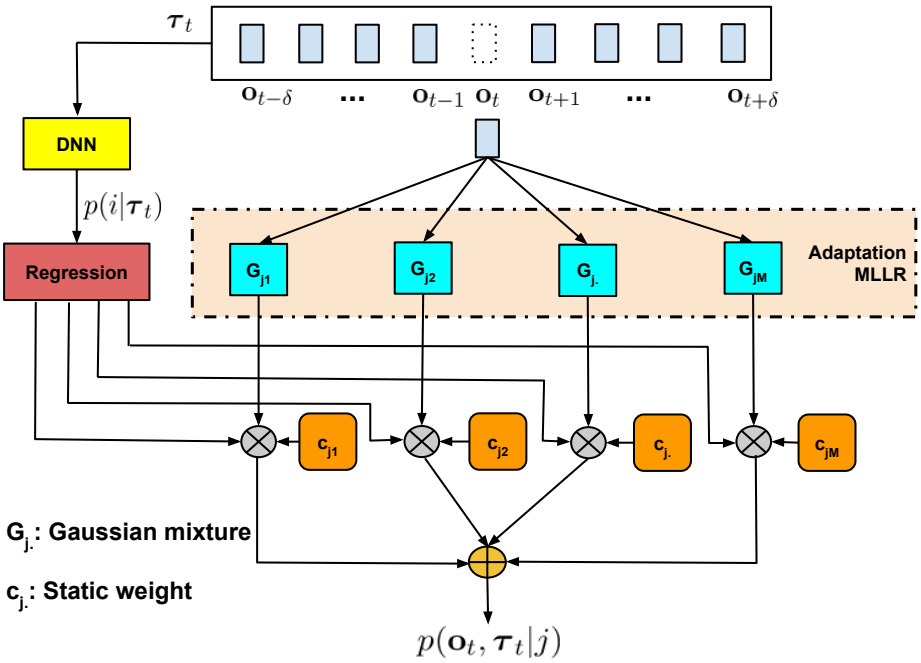

Figure 5.1: A schematic diagram showing the state output probability function of the proposed GMM+DNN/HMM system.

5.3 Regression of CD-DNN Posteriors

In order to maintain a reasonable number of regression parameters, the high-dimensional CD posterior features have to be *projected* down to a lower dimension. In the following, two solutions will be presented to achieve this. The first solution attempts to reduce the number of regression weights via parameter tying. The latent variables, i, are clustered into groups so that Eq-5.4 can be rewritten as:

$$p(\boldsymbol{\xi}_t|j) \propto \sum_{m=1}^{M} c_{jm} \sum_{g=1}^{G} \tilde{p}(g|\boldsymbol{\tau}_t) P(g|j,m) p(\mathbf{o}_t|j,m) \tag{5.9}$$

where $P(g|j,m)$ is the regression weight for group g and the normalized posterior probability of g is given by:

$$\tilde{p}(g|\boldsymbol{\tau}_t) = \frac{p(g|\boldsymbol{\tau}_t)}{P(g)} = \frac{\sum_{i \in g} p(i|\boldsymbol{\tau}_t)}{P(g)} \tag{5.10}$$

Therefore, tying the regression weights into groups is equivalent to projecting the posterior probabilities according to Eq-5.10. In this work, the groupings are chosen to correspond to the monophone states and the resulting group posteriors are simply the CI posteriors. However, such projection operation may lose valuable context information such that we no longer preserve the superior performance of the CD-DNN model. In order to incorporate

richer context information without dramatically increasing the model complexity, multi-stream TVWR [13] is used to integrate multiple sets of posterior features. The resulting state probability is given as:

$$p(\boldsymbol{\xi}_t|j) \propto \sum_{m=1}^{M} c_{jm} \prod_{c=1}^{C} \sum_{g_c=1}^{N_c} \tilde{p}(g_c|\boldsymbol{\tau}_t) p(g_c|j,m) p(\mathbf{o}_t|j,m)$$

where the CD posteriors are now factorized into C groupings and N_c is the number of groups in the c^{th} stream. In this work, we used three streams, one for the centre monophone states (g_2) and the other two for the left (g_1) and right (g_3) contexts. Although multiple DNNs can be trained for each stream, temporal context expansion can be applied to obtain the left and right context stream. Therefore, the posterior probabilities of the left/right contexts are derived from the centre-phone state:

$$p(g_1|\boldsymbol{\tau}_t) = p(g_2|\boldsymbol{\tau}_{t-\Delta_l}) \quad \text{and} \quad p(g_3|\boldsymbol{\tau}_t) = p(g_2|\boldsymbol{\tau}_{t+\Delta_r})$$

where Δ_l and Δ_r are the smallest positive values such that the states corresponding to the largest $p(g_2|\boldsymbol{\tau}_{t-\Delta_l})$ and $p(g_2|\boldsymbol{\tau}_{t+\Delta_r})$ are different from state with largest $p(g_2|\boldsymbol{\tau}_t)$.

The second CD posterior projection method adopts a *sparse* regression model where only a smaller set of CD posteriors is used for each Gaussian component. Considering that the TVWR formulation has the constraint $\sum_{i=1}^{N} P(i|j,m) = 1$, many of the regression parameters could be very small, especially when N is large. Hence, there may be only a small fraction of the parameters that actually contribute to the regression. The objective is to perform the sparse regression using only the most important parameters:

$$p(\boldsymbol{\xi}_t|j) \propto \sum_{m=1}^{M} c_{jm} \sum_{i \in I_j} \tilde{p}(i|\boldsymbol{\tau}_t) P(i|j,m) p(\mathbf{o}_t|j,m) \tag{5.11}$$

where I_j denotes the set of *active* latent variables for state j. Intuitively, I_j can be chosen such that:

$$I_j = \{i : P(i|j) > \nu\} \tag{5.12}$$

where $P(i|j)$ is computed as

$$P(i|j) = \frac{1}{|T_j|} \sum_{t \in T_j} p(i|\boldsymbol{\tau}_t) \tag{5.13}$$

T_j is the set of frames where j is the reference state and $|.|$ denotes the cardinality. ν is a threshold that can be adjusted to control the number of active posteriors, which may be chosen as $\nu = 1/N$. The regression parameters can then be initialized as $P(i|j,m) = P(i|j)$ if $i \in I_j$ and zero otherwise.

5.4 Experimental Results

The experiments were conducted on the Topic Detection and Tracking - Phase 3 (TDT3) corpus for English broadcast news transcription. After pre-processing of the original data, which includes removing non-speech segments, text normalization and audio segmentation, approximately 100 hours of data were retained for system training. The evaluation task is a 58k open vocabulary broadcast news transcription task taken from the F0 portion of the 1997 Hub-4E Benchmark Test. This testing set consists of about 3 hours of speech and 49 speakers. The decoding language model was obtained by interpolating two language models trained on the TDT3 transcriptions and the Gigaword English corpus, respectively.

The acoustic features are the 39 dimensional PLP coefficients (12 static coefficients, an energy term and the first two derivatives) with utterance-based cepstral mean and variance normalization [1]. The GMM/HMM baseline system is a decision tree state clustered triphone system with 4451 tied states. Each triphone is modelled by a 3-state left-to-right HMM and each state is modelled by a 20-component GMM. All the GMM parameters were trained using the maximum likelihood criterion, while the DNN was based on cross-entropy training criterion. For the DNN fine-tuning, 10 hours of the training data were held out as the cross validation set and the rest were used for parameter estimation. The input to the DNN is 15-frame context expanded PLP features with total dimension 585. The DNN has 5 hidden layers and each hidden layer has 2048 units with sigmoid activation function. The output layer of the DNN corresponds to the 3052 tied states of another GMM/HMM system. The recognition [2] was performed with a bigram full decoding followed by a trigram lattice rescoring. To build the tandem system, principle component analysis (PCA) was used to project the CI log posterior features into 13-dimensional features, which were then appended to the 39-dimensional PLP features. Finally, the GMM/HMM system was estimated using 52-dimensional tandem features and maximum likelihood criterion. To perform the unsupervised adaptation, the speaker independent system was first used to generate the hypotheses. Given the recognized hypotheses, CMLLR adaptation using either a global transform or regression classes were evaluated in this experiments. Particularly, 32 regression classes were generated and the split threshold was set to 2000. The global CMLLR transform was also used to generate the speaker dependent feature for adapting DNN. In this experiment, the threshold used for sparse regression was 1/120 with CI posteriors, and 0.001 with CD posteriors.

Table. 5.1 shows the performance of various baseline systems with or without unsupervised speaker adaptation. Without adaptation, the CD-DNN/HMM system obtained 3.2% absolute (or 20% relative) WER reduction over the ML trained GMM/HMM system,

[1] The results in [112] were based on cepstral mean normalized acoustic features.

[2] The results in [112] did not consider the prior probability of the states during decoding, which has been fixed in this experiment.

System	Adaptation		
	None	MLLR	class
GMM/HMM	16.0	14.8	14.6
Tandem	16.4	14.7	14.3
CD-DNN/HMM	**12.8**	**12.1**	-

Table 5.1: *WER(%) of various baseline systems with or without unsupervised speaker adaptation.*

which is consistent with many other reports. Unfortunately, the tandem system degraded compared to the baseline GMM/HMM system, which might be because the information between the acoustic feature and the posterior features has a lot of inconsistency. After performing unsupervised speaker adaptation using a global transform, absolute WER reductions of 1.2% and 1.7% were obtained by the GMM/HMM and the tandem systems, respectively. Further minor improvements can also be obtained by performing regression class based adaptation. Eventually, the tandem system becomes marginally better than the GMM/HMM system. When the speaker adapted feature was used for DNN decoding, 0.7% absolute improvement was obtained. This shows that fMLLR approach is quite an effective approach for adapting DNN [1]. Hence, we suggest using TVWR to perform the combination in a more systematic fashion.

Posteriors	Context Expansion	Regression Parameters	w/ un-adapted DNN			w/ adapted DNN	
			None	MLLR	class	MLLR	class
CI	No	10	12.9	11.9	11.9	12.0	11.8
	Yes	30	11.9	11.5	11.3	**11.0**	**10.8**
CD	No	70	11.8	**11.2**	**11.0**	11.4	11.1
	Yes	210	**11.7**	11.3	**11.0**	11.1	**10.8**

Table 5.2: *Comparison of the number of regression parameters per Gaussian component and the WER (%) performance of various GMM+DNN/HMM systems with or without context expansion and unsupervised speaker adaptation.*

Table. 5.2 shows the performance of the GMM+DNN/HMM systems with different configurations. When using the CI posteriors without context expansion, the un-adapted TVWR system gives 12.9% WER, which is 0.1% behind the CD-DNN/HMM system. This is possible due to the information loss after posterior grouping. After performing context expansion, the GMM+DNN/HMM performance improved to 11.9%, which

[1]When comparing these results with the report [112], the degradation might be caused by another fact that different feature space was used for GMM (PLP+CMN) and DNN (PLP+CMVN).

is 0.8% better than the CD-DNN/HMM baseline. When CD posteriors were used, the GMM+DNN/HMM with or without context expansion has significantly outperformed the CD-DNN/HMM baseline system. These results showed the effectiveness of performing sparse regression by selecting state-dependent active posteriors. When un-adapted DNN posteriors were used, performing unsupervised speaker adaptation yields consistent improvements of 0.2% – 1.1% for all the GMM+DNN/HMM systems. Regression class based adaptation also helped to slightly improve the systems with context expansion. However, the improvements are quite minor which might be because the adaptation data for each speaker is not enough. Anyway, these results show that the GMM+DNN/HMM systems are able to exploit the adaptability of the GMMs to obtain further improvements. After speaker adapted DNN posteriors were used, we have not seen consistent improvements for all systems. This might be because the adaptation techniques for GMM and DNN are in the same family so that less complementary information can be obtained. However, GMM+DNN/HMM with context-expanded adapted posteriors has shown consistent improvements due to the expansion of better posteriors. The best performance of 10.8% WER is obtained using the context expanded CD posteriors, which translates to 10.7% relative improvement compared to the speaker adapted DNN/HMM system. Although GMM+DNN/HMM using context expanded CD posteriors gives better performance than that using CI posteriors, this superiority has not been carried into the systems using adapted posterior features. This is probably because the rich information from the DNN posteriors has somewhat de-weighted the importance of the GMMs. As a result, there is less impact from applying MLLR adaptation to the GMMs.

Finally, we analyze the number of regression parameters per Gaussian component for the various GMM+DNN/HMM systems. If the raw CD posteriors are directly used, there will be 3052 regression weights per Gaussian component. By using different projection methods, the number of regression weights can be reduced to about an order magnitude smaller. It is worth noting that the preliminary results presented in this section have considered only the straightforward projection configurations. It may be possible to achieve further performance gains by adjusting the regression model complexity (*e.g.* using groupings other than the CI groups or using a different threshold, ν, for the sparse regression). It is also important to note that the threshold for sparse regression somehow controls the weights of generative model and discriminative model. A good balance between these two kinds of models may be achieved by tuning the threshold on a development set.

5.5 Summary

This chapter has proposed combining the GMM and DNN models using the Temporally Varying Weight Regression (TVWR) framework to achieve a high quality and adaptable state probability model for automatic speech recognition. The resulting GMM+DNN/HMM system is different from the tandem systems in that the GMMs are trained directly on the

cepstral acoustic features, rather than the DNN-derived tandem features. This chapter has focused on addressing the issue of incorporating the high dimensional CD-DNN state posteriors into the TVWR framework without dramatically increase the system complexity. Specifically, projected CI state posteriors, sparse regression and context expansion are introduced to mitigate the problem. Experimental results show that the proposed GMM+DNN/HMM system outperform the baseline DNN/HMM system. In additional, applying unsupervised speaker adaptation can further improve the performance of the proposed system. Future work will consider applying speaker adaptive training and discriminative training to the GMM+DNN/HMM system.

Chapter 6

Adaptation and Adaptive Training for Robust TVWR

In practice, speech data with the same transcription could have very different parameter representation due to various acoustic factors, such as background noises, transmission channels, speakers, etc. Hence, acoustic model trained on one condition may perform poorly on another condition. In this chapter, this issue is named as the robust speech recognition on mismatched conditions. In particular, noise and speaker will be the main focus of this chapter. Commonly used techniques for robust speech recognition are adaptation and adaptive training, including noise and speaker adaptation and adaptive training. The essence of adaptive training is to estimate a canonical acoustic model to be independent of various acoustic factors. In other words, speech irrelevant factors should be removed as many as possible from the final acoustic model during training, which is typically cooperated with respective adaptation techniques. During decoding, corresponding adaptation techniques can be applied to translate the canonical acoustic model to be condition dependent using small amount of adaptation data. Depending on the availability of correct transcription, adaptation can be either supervised or unsupervised. In order to improve the robustness against various testing conditions, adaptation and adaptive training for TVWR will be studied in this chapter. In details, robust TVWR with two configurations will be considered: one is using GMM as front-end to produce posterior features, another is using DNN as front-end. The advantage of using GMM as front-end is that various model compensation/adaptation approaches have been developed. Hence, compensating TVWR with GMM front-end is pretty straightforward except that the Gaussian parameters are corresponding to the long span acoustic features. This work has been published to the proceeding of Interspeech 2013 [120]. On the other hand, although DNN has difficulties to be compensated for robust speech recognition, multi-style trained DNN without adaptation has been found to outperform the GMM system with compensation. Therefore, it is interesting to take advantage of the best of two worlds using TVWR

framework. This work will serve as the second part of this chapter.

6.1 Robust TVWR using GMM based Posteriors

In this section, the Temporally Varying Weight Regression (TVWR) is investigated in two ways for noise robust speech recognition. Conventional model compensation approaches assume that the noise data is independent and identically distributed (i.i.d). Hence, non-stationary noise environments are poorly compensated using conventional model compensation approaches in the standard Hidden Markov Model (HMM) framework. TVWR maintains both the basic HMM structure and additional time-varying property, therefore, model compensation for TVWR can be performed to relax i.i.d. noise assumption. Second, although Noise Adaptive Training (NAT) [63] has been proposed to optimize the "pseudo-clean" HMM model for a better performance by maximizing the likelihood of multi-style data, NAT heavily depends on the simplicity of Vector Taylor Series (VTS) [58] formulation. Hence, other advanced compensation approaches, such as Trajectory-based Parallel Model Combination (TPMC) [61], have difficulties benefiting from this powerful training schema. This section will exploit the time-varying attribute of TVWR to approximate NAT such that any compensation technique can be applied during noise adaptive training. This work has been accepted to publish to proceeding of Interspeech 2013 [120].

6.1.1 Introduction

Noise robust speech recognition has been studied for many years but not solved yet due to its unsatisfactory accuracy for various noise conditions. One of the most popular approaches to solve this problem is to learn the impact of the noise on the corrupted speech such that the underlying acoustic model can be adapted to various unknown noisy environments. In order to apply the model compensation approaches such as [57, 58, 59, 60, 61], the noise model representing the noise evolution along with the time instance is expected. However, this noise model is typically modelled as a single state evolution process with single Gaussian model, which assumes that the noise feature is independent and identically distributed. Although this assumption simplifies both the noise model estimation process and the model adaptation formulae, it makes the non-stationary noise poorly modelled. Many techniques [121, 122, 123, 124] have been introduced to tackle the non-stationary noise problem, many of them have complicated formulations and require iterative estimation process. None of above approaches have solved this problem from the model compensation perspective: the adapted acoustic model works exactly the same as the clean model does in the clean environment.

On the other hand, Noise Adaptive Training NAT [63] has been proposed to optimize a "pseudo-clean" canonical HMM model by maximizing the likelihood of multi-condition

training data. Although promising improvements have been obtained by NAT, this powerful training schema strictly depends on the formulation of VTS adaptation [58] in order to derive tractable update formulae. This means that other advanced model compensation approaches such as Data-driven Parallel Model Combination (DPMC) [60] and Trajectory-based Parallel Model Combination (TPMC) [61] have difficulties in benefiting from this training schema, since it is very difficult to evaluate the derivative of the corrupted statistics with respect to clean statistics or noise statistics for the optimization process.

In this section, above two issues will be solved based on the temporally varying weight regression (TVWR) [11] framework. Since the temporal correlation of the speech is partially modelled into the time-varying GMM weights, equivalent temporal correlation of the noise feature will be required to be modelled for consistent compensation of TVWR model such that non-stationary noise environment can be better characterized. First, we will introduce how TVWR model can be adapted for handling the non-stationary noise environment. Second, NAT is factorized with a time-varying factor together with the conventionally compensated component output probability, the time-varying attribute of TVWR is exploited to approximate such time-varying factor such that complex compensation techniques, such as TPMC [61], can benefit from this powerful training schema.

6.1.2 Model Compensation for TVWR

In the following section, $\mathbf{x}_t$, $\mathbf{y}_t$, $\mathbf{n}_t$, $\mathbf{h}_t$ are defined as the clean, noisy, noise cepstral feature and channel distortion, respectively. In order to improve the noise robustness of TVWR, its formulation in Eq-3.3 needs to be compensated properly. For a quick reference, TVWR formulation in Eq-3.3 is re-written here:

$$p(\boldsymbol{\xi}_t|j) = \sum_{m=1}^{M} \underbrace{P(m|j)p(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m)}_{c_{jmt}} \underbrace{p(\mathbf{o}_t|j, m)}_{\mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_{jm},\boldsymbol{\Sigma}_{jm})} \tag{6.1}$$

where the conditional probability is factorized and approximated as:

$$p(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m) \approx K_t \sum_{i=1}^{N} \underbrace{\tilde{p}(i|\boldsymbol{\tau}_t)}_{\tilde{h}_{ti}} \underbrace{P(i|j, m)}_{w_{jmi}} \tag{6.2}$$

If TVWR system is trained by maximum likelihood, Gaussian parameters in $p(\mathbf{o}_t|j, m)$ or Eq-6.1 represent the distribution of the speech data. Given a noise model representing the acoustic condition, any conventional compensation approach [57, 58, 59, 60, 61] can be applied to adapt these Gaussian parameters, which is referred to as **back-end** compensation. On the other hand, posterior feature, $p(i|\boldsymbol{\tau}_t)$ in Eq-6.2 also depends on the acoustic features, which is condition dependent. The condition dependent normalization term contributes uninformative knowledge to discriminate different components or states,

so that it can be ignored even in a mismatch condition. As with the standard model compensation methods, $P(i|j,m)$, $P(m|j)$ are assumed to be unaffected by the channel and noise distortion. Therefore, the remaining problem is to generate condition dependent posterior feature, which is also referred to as **front-end** compensation.

6.1.2.1 Acoustic Model Compensation

Acoustic model compensation aims to obtain the noise condition dependent acoustic model representing the statistics of noisy speech. First, the clean acoustic model is defined as $\{\boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\}$ and noise-k model as $\{\boldsymbol{\mu}_n^k, \boldsymbol{\Sigma}_n^k, \boldsymbol{\mu}_h^k\}$. When the nonlinear function in Eq-2.104 is expanded at point $\{\boldsymbol{\mu}_{jm,0}, \boldsymbol{\mu}_{n,0}^k, \boldsymbol{\mu}_{h,0}^k\}$ and evaluated at point $\{\boldsymbol{\mu}_{jm}, \boldsymbol{\mu}_n^k, \boldsymbol{\mu}_h^k\}$, the noisy acoustic model can be obtained as:

$$\boldsymbol{\mu}_{jm}^k \approx \boldsymbol{\mu}_{jm,0}^k + \mathbf{G}_{jm}^k(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) + \mathbf{G}_{jm}^k(\boldsymbol{\mu}_h^k - \boldsymbol{\mu}_{h,0}^k) + \mathbf{F}_{jm}^k(\boldsymbol{\mu}_n^k - \boldsymbol{\mu}_{n,0}^k) \tag{6.3}$$

$$\boldsymbol{\Sigma}_{jm}^k \approx \mathbf{G}_{jm}^k \boldsymbol{\Sigma}_{jm} \mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^k \boldsymbol{\Sigma}_n^k \mathbf{F}_{jm}^{T,k} \tag{6.4}$$

where

$$\boldsymbol{\mu}_{jm,0}^k = \boldsymbol{\mu}_{jm,0} + \boldsymbol{\mu}_{h,0}^k + \mathbf{g}\left(\boldsymbol{\mu}_{n,0}^k - \boldsymbol{\mu}_{jm,0} - \boldsymbol{\mu}_{h,0}^k\right) \tag{6.5}$$

$$\mathbf{G}_{jm}^k = \mathbf{C}\,\mathrm{Diag}\left(\frac{1}{1+\exp\{\mathbf{C}^{-1}(\boldsymbol{\mu}_{n,0}^k - \boldsymbol{\mu}_{jm,0} - \boldsymbol{\mu}_{h,0}^k)\}}\right)\mathbf{C}^{-1} \tag{6.6}$$

$$\mathbf{F}_{jm}^k = \mathbf{I} - \mathbf{G}_{jm}^k \tag{6.7}$$

The advantage of forcing expansion point to be different from evaluation point is that when the clean acoustic model and noise model are not known, we can set the expansion point to be any possible value as the starting point and optimize the unknown evaluation point iteratively. However, if both clean acoustic model and noise model are known, the expansion point and evaluation point can be simply set to be the same, therefore the noisy acoustic model can be obtained as:

$$\boldsymbol{\mu}_{jm}^k \approx \boldsymbol{\mu}_{jm} + \boldsymbol{\mu}_h^k + \mathbf{C}\log\left[1+\exp\{\mathbf{C}^{-1}(\boldsymbol{\mu}_n^k - \boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_h^k)\}\right] \tag{6.8}$$

$$\boldsymbol{\mu}_{\Delta jm}^k \approx \mathbf{G}_{jm}^k \boldsymbol{\mu}_{\Delta jm} \tag{6.9}$$

$$\boldsymbol{\mu}_{\Delta^2 jm}^k \approx \mathbf{G}_{jm}^k \boldsymbol{\mu}_{\Delta^2 jm} \tag{6.10}$$

$$\boldsymbol{\Sigma}_{jm}^k \approx \mathbf{G}_{jm}^k \boldsymbol{\Sigma}_{jm} \mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^k \boldsymbol{\Sigma}_n^k \mathbf{F}_{jm}^{T,k} \tag{6.11}$$

$$\boldsymbol{\Sigma}_{\Delta jm}^k \approx \mathbf{G}_{jm}^k \boldsymbol{\Sigma}_{\Delta jm} \mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^k \boldsymbol{\Sigma}_{\Delta n}^k \mathbf{F}_{jm}^{T,k} \tag{6.12}$$

$$\boldsymbol{\Sigma}_{\Delta^2 jm}^k \approx \mathbf{G}_{jm}^k \boldsymbol{\Sigma}_{\Delta^2 jm} \mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^k \boldsymbol{\Sigma}_{\Delta^2 n}^k \mathbf{F}_{jm}^{T,k} \tag{6.13}$$

As can be seen, the back-end compensation for TVWR is exactly the same as conventional model compensation for HMM system.

6.1.2.2 Posterior Synthesizer Compensation

Next, noise compensated posterior features are needed to be generated for robust TVWR. Before that, the respective long span of clean, noisy, noise and channel features are denoted as:

$$\boldsymbol{\chi}_t = \{\mathbf{x}_{t-\delta}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+\delta}\} \tag{6.14}$$
$$\boldsymbol{\psi}_t = \{\mathbf{y}_{t-\delta}, \ldots, \mathbf{y}_{t-1}, \mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+\delta}\} \tag{6.15}$$
$$\boldsymbol{\eta}_t = \{\mathbf{n}_{t-\delta}, \ldots, \mathbf{n}_{t-1}, \mathbf{n}_{t+1}, \ldots, \mathbf{n}_{t+\delta}\} \tag{6.16}$$
$$\boldsymbol{\rho}_t = \{\mathbf{h}_{t-\delta}, \ldots, \mathbf{h}_{t-1}, \mathbf{h}_{t+1}, \ldots, \mathbf{h}_{t+\delta}\} \tag{6.17}$$

In the original TVWR [11] work, the posterior feature is generated using a neural network for better accuracy. However, it is difficult to directly adapt the neural network or any other discriminate models [125], as their parameters do not represent the statistics of the features. To circumvent this problem, generative models using long span features are introduced:

$$p(i|\boldsymbol{\chi}_t) = \frac{\mathcal{N}(\boldsymbol{\chi}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)P(i)}{\sum_{i=1}^{N} \mathcal{N}(\boldsymbol{\chi}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)P(i)} \tag{6.18}$$

where $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$ represents the clean generative model for latent variable i. In order to obtain the condition dependent posterior feature, model compensation for these clean generative models can be performed. Given the long span noise model $\{\boldsymbol{\mu}_\eta^k, \boldsymbol{\Sigma}_\eta^k\}$ for condition k, the noisy speech model $\{\boldsymbol{\mu}_\psi^{ik}, \boldsymbol{\Sigma}_\psi^{ik}\}$ for model i at condition k can be obtained using extended VTS [59]:

$$\boldsymbol{\mu}_i^k \approx \boldsymbol{\mu}_i + \boldsymbol{\mu}_\rho^k + \mathbf{Q}\log\left[1 + \exp\{\mathbf{Q}^{-1}(\boldsymbol{\mu}_\eta^k - \boldsymbol{\mu}_i - \boldsymbol{\mu}_\rho^k)\}\right] \tag{6.19}$$
$$\boldsymbol{\Sigma}_i^k \approx \mathbf{J}_i^k \boldsymbol{\Sigma}_i \mathbf{J}_i^{T,k} + \mathbf{L}_i^k \boldsymbol{\Sigma}_\eta^k \mathbf{L}_i^{T,k} \tag{6.20}$$

where the channel distortion is assumed to be stationary, hence $\boldsymbol{\Sigma}_\rho^k = \mathbf{0}$, and

$$\mathbf{Q} = \mathbf{I}_{2\delta} \otimes \mathbf{C} \tag{6.21}$$
$$\mathbf{J}_i^k = \mathbf{Q}\,\mathrm{Diag}\left\{\frac{1}{1 + \exp\{\mathbf{Q}^{-1}(\boldsymbol{\mu}_\eta^k - \boldsymbol{\mu}_i - \boldsymbol{\mu}_\rho^k)\}}\right\}\mathbf{Q}^{-1} \tag{6.22}$$
$$\mathbf{L}_i^k = \mathbf{I} - \mathbf{J}_i^k \tag{6.23}$$

$\mathbf{I}_{2\delta}$ is an identity matrix with dimension 2δ, $\mathbf{C}$ and $\mathbf{C}^{-1}$ are the DCT matrix and its pseudo-inverse matrix, $diag\{.\}$ denotes a diagonal matrix using the input vector element as its diagonal element. Based on above equations, block-wise (a.k.a. frame-wise) covariance matrix at index p and q can be re-written as:

$$\boldsymbol{\Sigma}_{i,pq}^k = \mathbf{J}_{i,p}^k \boldsymbol{\Sigma}_{i,pq} \mathbf{J}_{i,q}^{T,k} + \mathbf{L}_{i,p}^k \boldsymbol{\Sigma}_{\eta,pq}^k \mathbf{L}_{ik,q}^{T,k} \tag{6.24}$$

where $\mathbf{J}_{i,p}^{k}$ $\mathbf{J}_{i,q}^{k}$ indicates the diagonal block of $\mathbf{J}_{i}^{k}$ at index p and q respectively, and the block index ranges: $p, q \in [1, 2\delta]$. Conventional model compensation approaches in HMM framework assume that:

$$\boldsymbol{\Sigma}_{\eta,pq}^{k} = \mathbf{0} \qquad \text{if } p \neq q \tag{6.25}$$

$$\boldsymbol{\Sigma}_{\eta,pp}^{k} = \boldsymbol{\Sigma}_{\eta,qq}^{k} \quad \forall p, q \tag{6.26}$$

which is poorly made for non-stationary noise environment. In order to compensate TVWR model, the correlation of successive noise data is explicitly modelled into $\boldsymbol{\Sigma}_{\eta,pq}^{k}$ and compensated into the statistics $\boldsymbol{\Sigma}_{i,pq}^{k}$ of the corrupted speech, the i.i.d noise assumption can be relaxed. Compared to the back-end compensation, front-end compensation will compensate and maintain full covariance matrix in order to model both the intra-frame and inter-frame correlation. Since full covariance matrix is applied, it is impossible to cooperate the dynamic parameters due to its linear relationship to the static parameters.

6.1.3 NAT Approximation using TVWR

In the recently proposed noise adaptive training NAT [63] technique, both the "pseudo-clean" speech and noise models are optimized to maximize the likelihood of the multi-condition data given the condition adapted noisy speech models. The state emission probability for condition k after NAT on HMM system can be expressed as:

$$p(\mathbf{y}_t^k|j,k) = \sum_{m=1}^{M} c_{jm}^{NAT} \mathcal{N}(\mathbf{y}_t^k; \boldsymbol{\mu}_{jm}^{k,NAT}, \boldsymbol{\Sigma}_{jm}^{k,NAT}) \tag{6.27}$$

$$= \sum_{m=1}^{M} z_{kjmt}^{NAT} \mathcal{N}(\mathbf{y}_t^k; \boldsymbol{\mu}_{jm}^{k}, \boldsymbol{\Sigma}_{jm}^{k}) \tag{6.28}$$

where c_{jm}^{NAT}, $\boldsymbol{\mu}_{jm}^{k,NAT}$, $\boldsymbol{\Sigma}_{jm}^{k,NAT}$ are NAT optimized weights and compensated speech model for condition k, $\boldsymbol{\mu}_{jm}^{k}$, $\boldsymbol{\Sigma}_{jm}^{k}$ are the conventionally compensated speech model without NAT, z_{kjmt}^{NAT} is the time-varying scaling factor to make the above equation hold. From the perspective of Eq-6.28, the existence of non-constant z_{kjmt}^{NAT} after NAT tells that the conventional compensation does not yield maximum likelihood of the condition specific noisy speeches. Since the auxiliary function of NAT is a highly nonlinear function of the variance variable, the variance update requires gradient or Newton's method, which somehow complicates the optimization process [63].

Similarly, the state emission probability for TVWR after adaptation to condition k can be expressed as:

$$p(\mathbf{y}_t^k, \boldsymbol{\psi}_t^k|j,k) = \sum_{m=1}^{M} c_{kjmt} \mathcal{N}(\mathbf{y}_t^k; \boldsymbol{\mu}_{jm}^{k}, \boldsymbol{\Sigma}_{jm}^{k}) \tag{6.29}$$

where c_{kjmt} is the time-varying weight adapted to condition k, $\boldsymbol{\psi}_t^k$ is the long span noisy context from condition k. When comparing Eq-6.28 with Eq-6.29, both NAT and TVWR share some similar structure but differ in the underlying formulation and training process. Since the superiority of NAT over the conventional compensation is the time-varying factor, z_{kjmt}^{NAT}, noise adaptive training for c_{kjmt} becomes an important step to approximate NAT using TVWR formulation. After switching to this approximation, any model compensation during noise adaptive training can be applied, as the speech (Gaussian parameter part) and noise models do not change iteratively. The EM auxiliary function of noise adaptive training for TVWR parameters can be defined as:

$$\mathcal{Q} = \sum_{k,t,j,m} \gamma_{jm}^{k}(t) \log c_{kjmt}^{NAT} \tag{6.30}$$

where $\gamma_{jm}^{k}(t)$ is the component occupancy given the current TVWR model after adaptation to condition k. Given approximations in Eq-3.4 and Eq-3.7, a new auxiliary function w.r.t. w_{jmi}^{NAT}, c_{jm}^{NAT} can be derived based on EM algorithm:

$$\begin{aligned}\mathcal{Q} &= \sum_{k,t,j,m} \gamma_{jm}^{k}(t) \log \left(c_{jm}^{NAT} \sum_{i} w_{jmi}^{NAT} p(\boldsymbol{\psi}_t^k | i, j, m, k) \right) \\ &\geq \sum_{k,t,j,m,i} \gamma_{jmi}^{k}(t) \left(\log w_{jmi}^{NAT} + \log p(\boldsymbol{\psi}_t^k | i, k) \right) + \sum_{k,t,j,m} \gamma_{jm}^{k}(t) \log c_{jm}^{NAT}\end{aligned} \tag{6.31}$$

where

$$\gamma_{jmi}^{k}(t) = \gamma_{jm}^{k}(t) \frac{\hat{w}_{jmi}^{NAT} p(\boldsymbol{\psi}_t^k | i, j, m, k, \hat{\lambda})}{\sum_i \hat{w}_{jmi}^{NAT} p(\boldsymbol{\psi}_t^k | i, j, m, k, \hat{\lambda})} \tag{6.32}$$

$$\approx \gamma_{jm}^{k}(t) \frac{\hat{w}_{jmi}^{NAT} p(\boldsymbol{\psi}_t^k | i, k, \hat{\lambda})}{\sum_i \hat{w}_{jmi}^{NAT} p(\boldsymbol{\psi}_t^k | i, k, \hat{\lambda})} \tag{6.33}$$

$$= \gamma_{jm}^{k}(t) \frac{\hat{w}_{jmi}^{NAT} \tilde{h}_{ti}^{k}}{\sum_i \hat{w}_{jmi}^{NAT} \tilde{h}_{ti}^{k}} \tag{6.34}$$

$\hat{w}_{jmi}^{NAT}$, $\hat{\lambda}$ are the current TVWR model parameters, and $p(\boldsymbol{\psi}_t^k | i, k)$ is component independent of w_{jmi}^{NAT} and c_{jm}^{NAT}. Closed form update formulae can be found using Lagrange Multiplier method such that:

$$c_{jm}^{NAT} = \frac{\sum_{k,t} \gamma_{jm}^{k}(t)}{\sum_{m,k,t} \gamma_{jm}^{k}(t)} \tag{6.35}$$

$$w_{jmi}^{NAT} = \frac{\sum_{k,t} \gamma_{jmi}^{k}(t)}{\sum_{i,k,t} \gamma_{jmi}^{k}(t)} \tag{6.36}$$

In summary, the noise adaptive training approximation using TVWR can be described as follows:

1. Initialize TVWR with $w_{jmi} = 1/N$ from a clean or multi-condition standard HMM system; estimate generative models for posterior features using the same training data.

2. Estimate regular sized and long span noise models for each condition using the head and tail frames of all related utterances.

3. Perform model compensation for TVWR.

4. Calculate $\gamma_{jm}^{k}(t)$ and $\gamma_{jmi}^{k}(t)$ to accumulate sufficient statistics given each condition compensated TVWR system.

5. Update c_{jm}^{NAT} and w_{jmi}^{NAT} using Eq-6.35 and Eq-6.36.

6. If the auxiliary function in Eq-6.31 converged or maximum iteration reached, exit; otherwise, increase iteration number and goto step 4.

As can be seen, there is no restriction about the model compensation approach during the above training process. Since Gaussian parameters in the above training process are not optimized to be "pseudo-clean", the possible accuracy loss by using the initial Gaussian parameters will be compensated by adaptive training regression parameters. Note that the power of above training schema heavily benefits from the large number of regression parameters, which is about half of that of Gaussian parameters. Compared to the complicated update formulae of Gaussian parameters in the standard NAT formulation [63], update formulae for TVWR regression parameters are much simpler and easier. Although above training process assumes that the condition independent Gaussian parameters is unchanged, these Gaussian parameters may be updated in step 5 just using the standard HMM update formulae with a better forward-backward alignment. On the other hand, if VTS compensation is applied, the "pseudo-clean" speech and regular sized noise model parameters may also be updated just following the standard NAT procedure [63].

6.1.4 Experimental Results

In this section, experiments are conducted on the Aurora 4 [104] corpus, which contains 3 datasets: clean dataset, multi-condition dataset, multi-noise dataset. Each dataset includes about 14 hours speech utterances, or 7138 utterances. Since only additive noise is considered in this section, the multi-condition dataset with channel distortion will not be used. In multi-noise training dataset, 6 noises are artificially added at 15 dB average SNR. For evaluation, only the first 7 test sets from the same microphone as the training data are used. For convenient presentation, Set-A and Set-B are introduced: Set-A contains 1 clean test set while Set-B contains 6 noisy test set with the same noises as multi-noise training data but different 10 dB average SNR. Each test set has the same 330 utterances for 5k closed vocabulary recognition task (NIST Nov'92 WSJ0).

The clean baseline HMM system is a decision tree state-clustered triphone system with 3226 distinct states and 16 components per state. The acoustic features are 39 dimensional MFCC [126], including 12 static coefficients, zero-th coefficient and the first two derivatives. Given the clean baseline system, multi-noise HMM system was estimated by one iteration of single pass retraining. Monophone posterior features, h_{ti} were generated from multiple GMM models: each monophone has 8 full-covariance mixture components, which may be estimated from the clean or multi-noise data. The context expanded features used here include a sequence of MFCC features spanning a window of 8 static features, i.e. 13*8 dimensions. The model compensation approaches investigated here include VTS [58] and Trajectory PMC (TPMC) [61], where TPMC has been shown to be better than VTS. To perform the noise adaptive training, only multi-noise training data are used. While the conventional NAT re-estimated both the "pseudo-clean" speech and noise models using VTS, NAT approximation using TVWR (NA-TVWR starting from multi-noise HMM) only updated the regression parameters and static weights(NA-TVWR* updated condition independent Gaussian parameters using the standard HMM update formulae). In the decoding, noise models are initially estimated by the head and tail (20) observations from each condition. To evaluate model compensation on the clean model, these initial noise models are always used without re-estimation for a fair comparison to TPMC. When evaluating VTS-based NA-TVWR and NAT-HMM, these noise models are re-estimated iteratively using multiple (2) iteration recognized hypotheses. The recognition results below includes a bigram full decoding followed by a trigram lattice-rescoring using HTK [5].

System	Adaptation	Set-A	Set-B	Avg
HMM	-	6.4	51.0	44.7
TVWR	-	5.8	56.5	49.3
HMM	VTS	7.8	18.2	16.7
TVWR		**6.8**	**16.0**	**14.7**
HMM	TPMC	6.9	14.8	13.6
TVWR		**6.2**	**12.3**	**11.4**

Table 6.1: *WER(%) for different approaches using clean training data.*

The recognition results of various approaches using different training data are shown in Table. 6.1 and Table. 6.2, where the "Avg" column indicates the average performance of all mentioned 7 test sets. When all systems are estimated using the clean data, both the standard HMM and TVWR systems without compensation suffered from dramatical performance degradation in the noisy test conditions. After performing model compensation, all the systems significantly outperformed the baseline systems. Compared to the standard HMM system using either VTS or TPMC, both adapted TVWR systems shows

Training	Adaptation	Set-A	Set-B	Avg
Mult-HMM	-	8.5	18.7	17.2
NAT-HMM	VTS	7.3	16.0	14.8
NA-TVWR	VTS	8.1	14.9	13.9
NA-TVWR	TPMC	7.9	12.5	11.8
NA-TVWR*	TPMC	**7.2**	**11.8**	**11.1**

Table 6.2: *WER(%) for different approaches using multi-noise training data.*

about 2% absolute average improvements. Since most testing speeches are corrupted by non-stationary noise, modelling the temporal correlation of the noise feature makes TVWR gain more than the standard HMM system. On the other hand, although more accurate corrupted dynamic parameters (delta, delta-delta) were estimated by TPMC for a better representation of the impact of non-stationary noise on the speech, TVWR can still obtain significant improvement. This tells that some characteristics of non-stationary noise corrupted speeches cannot be modelled well in the standard HMM framework.

When noise adaptive training on the multi-noise data was performed, NAT-HMM+VTS obtained more than 2% absolute improvement over the multi-noise HMM. NAT-HMM+VTS achieved better recognition accuracy on the clean test Set-A, probably because its "pseudo-clean" speech model has been optimized while TVWR still use the multi-noise estimated Gaussian parameters. However, our approximation approach, NA-TVWR+VTS achieved 0.9% absolute more average improvement over NAT-HMM+VTS. These results show that adaptively trained time-varying weight is able to approximate some property of NAT, while the additional gain from TVWR over NAT is probably due to the useful temporal correlation of noise features. When NA-TVWR was performed with TPMC, NA-TVWR+TPMC averagely outperformed other VTS-based systems. Although NA-TVWR+TPMC still is not as good as NAT-HMM+VTS at test Set-A due to using the multi-noise data estimated Gaussian parameters, NA-TVWR+TPMC performs quite well averagely and comparable to TVWR+TPMC trained by the clean data. After the condition independent Gaussian parameters were optimized just using the standard HMM update formulae, NA-TVWR*+TPMC performed best and also slightly better than the TPMC-based clean TVWR system. According to these results, any other compensation approach can potentially perform quite well given the multi-condition data in the TVWR framework.

6.1.5 Summary

In this section, proposed Temporally Varying Weight Regression (TVWR) model is investigated in two ways to perform noisy speech recognition. Firstly, model compensation

for TVWR is proposed such that i.i.d. noise assumption can be relaxed, which can potentially improve the accuracy in the non-stationary noise environment. Secondly, instead of directly updating the model and noise parameters in the standard NAT procedure, the time-varying attribute of TVWR is used to approximate the NAT algorithm such that any compensation approach can benefit from the multi-condition data training. Experimental results show that promising improvements can be obtained by performing model-based compensation for TVWR. Besides, the proposed NAT approximation technique achieved significant improvement over the traditional NAT using VTS compensation. This approximation can benefit more complicated compensation technique, such as Trajectory-based PMC, where efficient NAT algorithm cannot be derived.

6.2 Robust TVWR using DNN based Posteriors

Context-dependent Deep Neural Network has obtained consistent and significant improvements over the Gaussian Mixture Model (GMM) based systems for various speech recognition tasks. However, since DNN is discriminatively trained, it is more sensitive to label errors and not reliable for unsupervised adaptation. On the other hand, DNN parameters do not have a clear and meaningful interpretation, therefore, it has been difficult to develop effective adaptation techniques for the DNNs. Nevertheless, DNN without any adaptation has already shown superior performance to the GMM system after iteratively joint noise/speaker adaptation and adaptive training. Recently, Temporally Varying Weight Regression (TVWR) has been successfully applied to combine DNN and GMM for robust unsupervised speaker adaptation. In this section, joint speaker/noise adaptation and adaptive training of TVWR using DNN posteriors are investigated for robust speech recognition.

6.2.1 Introduction

In practice, many speech variabilities exist due to the acoustic environments, reverberations, background noise, communication channels, speaker difference. Training an acoustic model to be robust against these variabilities has been always important and challenging. When the Gaussian Mixture Model (GMM) was a dominant observation distribution representation of the Hidden Markov Model (HMM) stats, many techniques were developed to make it robust to various acoustic conditions. Since the GMM is a generative model whose parameters represent the statistics of the data, it is easier to manipulate the parameters to adapt to a specific condition. For example, Maximum a Posterior (MAP) [45] and Maximum Likelihood Linear Regression (MLLR) [48] are commonly used for speaker adaptation. On the other hand, Vector Taylor Series (VTS) is widely used to compensate acoustic features or models for different noise conditions. All these methods modify

the GMM parameters to better represent the data statistics in the respective acoustic conditions.

The objective function of DNN optimization is nonlinear, non-convex, and has no closed-form solution [1]. Hence, its weights are obtained by the typical line search method, and it is usually implemented by the error back-propagation algorithm. Parameter interpolation like MAP [45] does not work for the DNN weights. First, MAP estimation is not tractable for DNN training as DNN is not a generative model. Second, assuming that the objective function is to minimize the cross-entropy, as it is not convex, an interpolation of two local minimums may lead to a local maximum. On the other hand, linear regression like MLLR [48] is equivalent to adding a linear activation layer before each nonlinear (such as sigmoid) activation layer in DNN. As the number of linear regression parameters can be million in a state-of-the-art DNN based speech recognition system, robust estimation of those adaptation parameters is difficult for limited adaptation data. Furthermore, it is also difficult to control the complexity of the linear regression according to the amount of available adaptation data. Discriminatively learning a speaker code has been proposed to fast adapt a DNN based system [114]. In this work, adapting NN with speaker code serves as the speaker dependent transform. In order to achieve training two separate NNs: adapting NN and original NN, adaptive training is required. This may not work if training transcription does not contain speaker information. As speaker code is learned discriminatively, unsupervised adaptation may not be robust. Different from speaker variabilities, noise/channel has more serious impacts on the front-end parameterization or back-end acoustic model. Therefore, speak code based approach may not work for noise robustness.

Currently, the techniques that improve the noise robustness of the DNN-based acoustic models can be divided into two categories: multi-style training [127] and feature engineering [51, 76, 128, 129, 130]. Multi-style training aims to use multi-style training data, hopefully to learn as many speech variabilities as possible. Different from adaptive training that aims to remove variabilities from the final model, multi-style training learns those variabilities using a single model, which is very difficult. Eventually, the average performance among various conditions can be greatly improved but it sometimes degrades the performance on the clean environment. As multi-style training is pretty efficient for both traing and decoding, it has been widely used for the applications with many working conditions. On the other hand, feature engineering aims to either extract more robust acoustic features against variabilities or remove variabilities to get normalized acoustic features. If a DNN-based system is trained using the clean data, feature enhancement or denoising techniques, such as feature-based VTS [53, 128, 129] can be performed to cope with the noisy testing environment. If the stereo training data (a parallel clean and noisy dataset) is available, SPLICE [52], MMSE [51] or DRADE (Deep Recurrent Denoising Autoencoder) [130] can be applied to estimate the "clean" speech. In practice, it is much more expensive to prepare the stereo training data than the multi-style training data.

Recent work [127] has shown that the standard DNN system using filter-bank features and multi-style training achieved comparable performance to a GMM-based system with joint speaker and noise adaptive training [105]. Therefore, it is interesting to investigate whether combining these two systems will further improve the performance. In our recent work [112], Temporally Varying Weight Regression (TVWR) has been presented as an approach to combine GMM and DNN for speaker adaptation. VTS has also been applied to compensate TVWR using the GMM-based posterior features [120]. In this section, joint speaker/noise adaptation and adaptive training will be studied for robust TVWR using the DNN posteriors.

6.2.2 Noise Adaptation and Adaptive Training

In early sections of this chapter, TVWR has been formulated to approximate noise adaptive training such that various compensation approaches can benefit from the powerful noise adaptive training. Assumption has been made that noise data is available during training and decoding, and channel distortion is not considered. This is a practically poor assumption considering that noise data is not always available and channel distortion happens a lot. In order to circumvent this limitation, noise model (additive noise and channel distortion) estimation in the TVWR framework becomes critically important. On the other hand, model compensation approaches like VTS [58] assume the acoustic model (or Gaussian parameters) to be a "clean" generative model. When "clean" training data is not available, noise adaptive training (NAT) [63] has been proposed to estimate a "pseudo-clean" generative model. As noise adaptive training contains noise model and canonical acoustic model estimations, both models are estimated by maximizing the likelihood of multi-style training data (or the recognized hypothesis if estimating noise model during decoding). Given the posterior features predicted by a multi-style trained DNN, the "auxiliary" function for noise adaptive training in TVWR framework is given as:

$$\mathcal{Q} = \sum_{k,t,j,m} \gamma_{jm}^{k}(t) \log p(\mathbf{y}_t^k | j, m, k) \tag{6.37}$$

$$\begin{aligned} &= \sum_{k,t,j,m} \gamma_{jm}^{k}(t) \left(-\frac{1}{2} \log |\boldsymbol{\Sigma}_{jm}^{k}| - \frac{1}{2} (\mathbf{y}_t^k - \boldsymbol{\mu}_{jm}^{k})^T \boldsymbol{\Sigma}_{jm}^{-1,k} (\mathbf{y}_t^k - \boldsymbol{\mu}_{jm}^{k}) \right) \\ &+ \sum_{k,t,j,m} \gamma_{jm}^{k}(t) \left(-\frac{1}{2} \log |\boldsymbol{\Sigma}_{\Delta jm}^{k}| - \frac{1}{2} (\Delta \mathbf{y}_t^k - \boldsymbol{\mu}_{\Delta jm}^{k})^T \boldsymbol{\Sigma}_{\Delta jm}^{-1,k} (\Delta \mathbf{y}_t^k - \boldsymbol{\mu}_{\Delta jm}^{k}) \right) \\ &+ \sum_{k,t,j,m} \gamma_{jm}^{k}(t) \left(-\frac{1}{2} \log |\boldsymbol{\Sigma}_{\Delta^2 jm}^{k}| - \frac{1}{2} (\Delta^2 \mathbf{y}_t^k - \boldsymbol{\mu}_{\Delta^2 jm}^{k})^T \boldsymbol{\Sigma}_{\Delta^2 jm}^{-1,k} (\Delta^2 \mathbf{y}_t^k - \boldsymbol{\mu}_{\Delta^2 jm}^{k}) \right) \end{aligned} \tag{6.38}$$

where k represents the utterance or condition id, $\gamma_{jm}^{k}(t)$ is the component occupancy given the current adapted TVWR system using DNN based posterior features,

$$\gamma_{jm}^{k}(t) = \gamma_{j}^{k}(t)\frac{\tilde{c}_{jmt}(k,\hat{\mathbf{\Lambda}})\mathcal{N}(\mathbf{y}_{t}^{k};\hat{\boldsymbol{\mu}}_{jm}^{k},\hat{\mathbf{\Sigma}}_{jm}^{k})}{\sum_{m=1}^{M}\tilde{c}_{jmt}(k,\hat{\mathbf{\Lambda}})\mathcal{N}(\mathbf{y}_{t}^{k};\hat{\boldsymbol{\mu}}_{jm}^{k},\hat{\mathbf{\Sigma}}_{jm}^{k})} \tag{6.39}$$

$$= \gamma_{j}^{k}(t)\frac{\left(\sum_{i\in I_j}\hat{w}_{jmi}\tilde{p}(i|\boldsymbol{\chi}_{t}^{k})\right)\hat{c}_{jm}\mathcal{N}(\mathbf{y}_{t}^{k};\hat{\boldsymbol{\mu}}_{jm}^{k},\hat{\mathbf{\Sigma}}_{jm}^{k})}{\sum_{m=1}^{M}\left(\sum_{i\in I_j}\hat{w}_{jmi}\tilde{p}(i|\boldsymbol{\chi}_{t}^{k})\right)\hat{c}_{jm}\mathcal{N}(\mathbf{y}_{t}^{k};\hat{\boldsymbol{\mu}}_{jm}^{k},\hat{\mathbf{\Sigma}}_{jm}^{k})} \tag{6.40}$$

$\tilde{p}(i|\boldsymbol{\chi}_{t}^{k}) = p(i|\boldsymbol{\chi}_{t}^{k})/P(i)$ is the normalized posterior feature predicted by multi-style trained DNN (Hence, DNN is independent of k), $P(i)$ is the prior probability of label i (which may be assumed to be uniform or estimated from training data in case of bias labels), VTS expansion functions for hidden variable (noise model and acoustic model) estimation are given as:

$$\boldsymbol{\mu}_{jm}^{k} \approx \boldsymbol{\mu}_{jm,0}^{k} + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{h}^{k} - \boldsymbol{\mu}_{h,0}^{k}) + \mathbf{F}_{jm}^{k}(\boldsymbol{\mu}_{n}^{k} - \boldsymbol{\mu}_{n,0}^{k}) \tag{6.41}$$

$$\boldsymbol{\mu}_{\Delta jm}^{k} \approx \boldsymbol{\mu}_{\Delta jm,0}^{k} + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{\Delta jm} - \boldsymbol{\mu}_{\Delta jm,0}) \tag{6.42}$$

$$\boldsymbol{\mu}_{\Delta^2 jm}^{k} \approx \boldsymbol{\mu}_{\Delta^2 jm,0}^{k} + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{\Delta^2 jm} - \boldsymbol{\mu}_{\Delta^2 jm,0}) \tag{6.43}$$

$$\mathbf{\Sigma}_{jm}^{k} \approx \mathbf{G}_{jm}^{k}\mathbf{\Sigma}_{jm}\mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^{k}\mathbf{\Sigma}_{n}^{k}\mathbf{F}_{jm}^{T,k} \tag{6.44}$$

$$\mathbf{\Sigma}_{\Delta jm}^{k} \approx \mathbf{G}_{jm}^{k}\mathbf{\Sigma}_{\Delta jm}\mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^{k}\mathbf{\Sigma}_{\Delta n}^{k}\mathbf{F}_{jm}^{T,k} \tag{6.45}$$

$$\mathbf{\Sigma}_{\Delta^2 jm}^{k} \approx \mathbf{G}_{jm}^{k}\mathbf{\Sigma}_{\Delta^2 jm}\mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^{k}\mathbf{\Sigma}_{\Delta^2 n}^{k}\mathbf{F}_{jm}^{T,k} \tag{6.46}$$

compensated acoustic models based on the current parameters are given as:

$$\boldsymbol{\mu}_{jm,0}^{k} \approx \boldsymbol{\mu}_{jm,0} + \boldsymbol{\mu}_{h,0}^{k} + \mathbf{g}\left(\boldsymbol{\mu}_{n,0}^{k} - \boldsymbol{\mu}_{jm,0} - \boldsymbol{\mu}_{h,0}^{k}\right) \tag{6.47}$$

$$\boldsymbol{\mu}_{\Delta jm,0}^{k} \approx \mathbf{G}_{jm}^{k}\boldsymbol{\mu}_{\Delta jm,0} \tag{6.48}$$

$$\boldsymbol{\mu}_{\Delta^2 jm,0}^{k} \approx \mathbf{G}_{jm}^{k}\boldsymbol{\mu}_{\Delta^2 jm,0} \tag{6.49}$$

expansion Jacobian with respect to mean variables are given as:

$$\mathbf{G}_{jm}^{k} = \mathbf{C}\,\mathrm{Diag}\left(\frac{1}{1+\exp\{\mathbf{C}^{-1}(\boldsymbol{\mu}_{n,0}^{k} - \boldsymbol{\mu}_{jm,0} - \boldsymbol{\mu}_{h,0}^{k})\}}\right)\mathbf{C}^{-1} \tag{6.50}$$

$$\mathbf{F}_{jm}^{k} = \mathbf{I} - \mathbf{G}_{jm}^{k} \tag{6.51}$$

where the first/second additive and channel means of noise model are assumed to be zero, such that $\boldsymbol{\mu}_{\Delta h}^{k} = \boldsymbol{\mu}_{\Delta^2 h}^{k} = \boldsymbol{\mu}_{\Delta n}^{k} = \boldsymbol{\mu}_{\Delta^2 n}^{k} = \mathbf{0}$, subscript $_0$ represents the expansion point, which corresponds to the current model parameters (or initial model parameters if it is at the first iteration). Depending on the available training data, current acoustic model can be either clean or noisy.

6.2.2.1 Noise Model Estimation

In order to apply model compensation such as VTS [58], additive noise model (usually represented by a single Gaussian component) and channel distortion (usually assumed to be static) have to be prepared. If the channel distortion is assumed to be nonexistent, additive noise model may be rapidly estimated from a short recording of the background noise. Since there is no way to directly estimate channel distortion, estimation-maximization (EM) algorithm can be applied to estimate those hidden variables. In this case, noise model is estimated by maximizing the likelihood of the speech data given the transcription or hypothesis. The initial noise model may be estimated using head and tail frames of an utterance, e.g. 20 frames from each side, while channel mean is initialized to be zero, channel variance is assumed to be zero. Different from VTS adaptation, which assumes that the acoustic model should be clean, noise model estimation based on VTS formulation does not have such assumption. In other words, if the current acoustic model is noisy, the final estimate is no longer a "noise" model. Instead, this "noise" model just represents a transformation that maximizes the likelihood of the hypothesis. When performing noise model estimation, current acoustic model parameters are assumed to be fixed such that $\boldsymbol{\mu}_{jm} = \boldsymbol{\mu}_{jm,0}$, $\boldsymbol{\Sigma}_{jm} = \boldsymbol{\Sigma}_{jm,0}$. Hence, the compensated cepstral mean becomes:

$$\boldsymbol{\mu}_{jm}^{k} \approx \boldsymbol{\mu}_{jm,0}^{k} + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{h}^{k} - \boldsymbol{\mu}_{h,0}^{k}) + \mathbf{F}_{jm}^{k}(\boldsymbol{\mu}_{n}^{k} - \boldsymbol{\mu}_{n,0}^{k}) \tag{6.52}$$

Noise Mean Update In order to obtain the optimal estimation of additive noise mean and channel distortion, the partial derivatives with respect to $\boldsymbol{\mu}_{n}^{k}$ and $\boldsymbol{\mu}_{h}^{k}$ are taken respectively as:

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_{n}^{k}} = \sum_{t,j,m} \gamma_{jm}^{k}(t) \left(\frac{\partial \boldsymbol{\mu}_{jm}^{k}}{\partial \boldsymbol{\mu}_{n}^{k}}\bigg|_{\boldsymbol{\mu}_{n,0}}^{T} \boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_{t}^{k} - \boldsymbol{\mu}_{jm}^{k}) \right) \tag{6.53}$$

$$= \sum_{t,j,m} \gamma_{jm}^{k}(t) \left(\mathbf{F}_{jm}^{T,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \left(\mathbf{y}_{t}^{k} - \boldsymbol{\mu}_{jm,0}^{k} - \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{h}^{k} - \boldsymbol{\mu}_{h,0}^{k}) - \mathbf{F}_{jm}^{k}(\boldsymbol{\mu}_{n}^{k} - \boldsymbol{\mu}_{n,0}^{k}) \right) \right) \tag{6.54}$$

and

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_{h}^{k}} = \sum_{t,j,m} \gamma_{jm}^{k}(t) \left(\frac{\partial \boldsymbol{\mu}_{jm}^{k}}{\partial \boldsymbol{\mu}_{h}^{k}}\bigg|_{\boldsymbol{\mu}_{h,0}}^{T} \boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_{t}^{k} - \boldsymbol{\mu}_{jm}^{k}) \right) \tag{6.55}$$

$$= \sum_{t,j,m} \gamma_{jm}^{k}(t) \left(\mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \left(\mathbf{y}_{t}^{k} - \boldsymbol{\mu}_{jm,0}^{k} - \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{h}^{k} - \boldsymbol{\mu}_{h,0}^{k}) - \mathbf{F}_{jm}^{k}(\boldsymbol{\mu}_{n}^{k} - \boldsymbol{\mu}_{n,0}^{k}) \right) \right) \tag{6.56}$$

After some algebra manipulations and setting above derivatives to zero, an equation system can be obtained as:

$$\mathbf{A}\boldsymbol{\mu}_{h}^{k} + \mathbf{B}\boldsymbol{\mu}_{n}^{k} = \mathbf{g} \tag{6.57}$$

$$\mathbf{D}\boldsymbol{\mu}_{h}^{k} + \mathbf{K}\boldsymbol{\mu}_{n}^{k} = \mathbf{h} \tag{6.58}$$

where

$$\mathbf{A} = \sum_{t,j,m} \gamma_{jm}^{k}(t)\mathbf{F}_{jm}^{T,k}\boldsymbol{\Sigma}_{jm}^{-1,k}\mathbf{G}_{jm}^{k} \tag{6.59}$$

$$\mathbf{B} = \sum_{t,j,m} \gamma_{jm}^{k}(t)\mathbf{F}_{jm}^{T,k}\boldsymbol{\Sigma}_{jm}^{-1,k}\mathbf{F}_{jm}^{k} \tag{6.60}$$

$$\mathbf{D} = \sum_{t,j,m} \gamma_{jm}^{k}(t)\mathbf{G}_{jm}^{T,k}\boldsymbol{\Sigma}_{jm}^{-1,k}\mathbf{G}_{jm}^{k} \tag{6.61}$$

$$\mathbf{K} = \sum_{t,j,m} \gamma_{jm}^{k}(t)\mathbf{G}_{jm}^{T,k}\boldsymbol{\Sigma}_{jm}^{-1,k}\mathbf{F}_{jm}^{k} \tag{6.62}$$

$$\mathbf{g} = \sum_{t,j,m} \gamma_{jm}^{k}(t)\mathbf{F}_{jm}^{T,k}\boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_{t}^{k} - \boldsymbol{\mu}_{jm,0}^{k} + \mathbf{G}_{jm}^{k}\boldsymbol{\mu}_{h,0}^{k} + \mathbf{F}_{jm}^{k}\boldsymbol{\mu}_{n,0}^{k}) \tag{6.63}$$

$$\mathbf{h} = \sum_{t,j,m} \gamma_{jm}^{k}(t)\mathbf{G}_{jm}^{T,k}\boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_{t}^{k} - \boldsymbol{\mu}_{jm,0}^{k} + \mathbf{G}_{jm}^{k}\boldsymbol{\mu}_{h,0}^{k} + \mathbf{F}_{jm}^{k}\boldsymbol{\mu}_{n,0}^{k}) \tag{6.64}$$

Note that $\mathbf{A} = \mathbf{K}^{T}$. Solving above linear system can give the following closed form update formulae used for parameter estimation:

$$\boldsymbol{\mu}_{h}^{k} = (\mathbf{A} - \mathbf{B}\mathbf{K}^{-1}\mathbf{D})^{-1}(\mathbf{g} - \mathbf{B}\mathbf{K}^{-1}\mathbf{h}) \tag{6.65}$$

$$\boldsymbol{\mu}_{n}^{k} = (\mathbf{B} - \mathbf{A}\mathbf{D}^{-1}\mathbf{K})^{-1}(\mathbf{g} - \mathbf{A}\mathbf{D}^{-1}\mathbf{h}) \tag{6.66}$$

$$= (\mathbf{K} - \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}(\mathbf{h} - \mathbf{D}\mathbf{A}^{-1}\mathbf{g}) \tag{6.67}$$

The original NAT [63] estimates $\boldsymbol{\mu}_{h}^{k}$ by setting $\boldsymbol{\mu}_{n}^{k} = \boldsymbol{\mu}_{n,0}^{k}$, which may yield a simpler formula at the cost of slower convergence. In practice, an issue arises with above estimation procedure. Auxiliary function of log-likelihood function is evaluated based on the compensation function in Eq-6.8 (i.e. $\boldsymbol{\mu}_{n}^{k} = \boldsymbol{\mu}_{n,0}^{k}$), while "auxiliary" function for noise model estimation uses the compensation function in Eq-6.52 (i.e. $\boldsymbol{\mu}_{n}^{k} \neq \boldsymbol{\mu}_{n,0}^{k}$). Hence, increasing the "auxiliary" function for noise model estimation does not guarantee the increase of final log-likelihood. Assuming that the forward-backward alignment is fixed, the auxiliary function based on compensation function in Eq-6.8 can be quickly evaluated for the new noise model estimate. The target is to ensure the following inequality:

$$\mathcal{Q}(\boldsymbol{\mu}_{n,0}^{k} = \phi_{new}^{k}) \geq \mathcal{Q}(\boldsymbol{\mu}_{n,0}^{k} = \phi_{initial}^{k}) \tag{6.68}$$

Otherwise, we can back-off the update by interpolating the initial and update parameters such as:

$$\phi_{new}^{k} = \lambda\phi_{update}^{k} + (1-\lambda)\phi_{initial}^{k} \tag{6.69}$$

where the interpolation λ is iteratively reduced by a factor of 2 till the inequality in IEq-6.68 holds. Since there is no need to go through the whole training data, this back-off process is very fast.

Noise Variance Update When updating noise variances, additive noise mean and channel mean are assumed to be fixed, such that $\boldsymbol{\mu}_n^k = \boldsymbol{\mu}_{n,0}^k$, $\boldsymbol{\mu}_h^k = \boldsymbol{\mu}_{h,0}^k$. In addition, all covariance matrices are assumed to be diagonal. In case of updating the variance of D-dimension static parameters, the auxiliary function with respect to utterance k noise parameters can be re-written as:

$$\mathcal{Q} = \sum_{t,j,m} -\frac{1}{2}\gamma_{jm}^k(t)\left\{\sum_{d=1}^{D}\left(\log\sigma_{jm,d}^{2,k} + \frac{(y_{t,d}^k - \mu_{jm,d}^k)^2}{\sigma_{jm,d}^{2,k}}\right)\right\} \tag{6.70}$$

where

$$\sigma_{jm,d}^{2,k} = \sum_{i=1}^{D} g_{jmk,di}^2\sigma_{jm,i}^2 + f_{jmk,di}^2\sigma_{n,i}^{2,k}, \qquad d = 1,\ldots,D \tag{6.71}$$

$g_{jmk,di}$ and $f_{jmk,di}$ are the element located at row-d and column-i of $\mathbf{G}_{jm}^k$ and $\mathbf{F}_{jm}^k$, respectively. Instead of directly optimizing the target variable $\sigma_{n,p}^{2,k}$, the logarithm of the variance is estimated to ensure the positivity, therefore, an intermedia variable is defined as:

$$\tilde{\sigma}_{n,p}^{2,k} = \log\left(\sigma_{n,p}^{2,k}\right) \tag{6.72}$$

After $\tilde{\sigma}_{n,p}^{2,k}$ is estimated, the original variable can be obtained by:

$$\sigma_{n,p}^{2,k} = \exp\left(\tilde{\sigma}_{n,p}^{2,k}\right) \tag{6.73}$$

In order to optimize the noise variance, Newton's method is adopted, which needs up to second order of derivatives with respect to $\tilde{\sigma}_{n,p}^{2,k}$:

$$\frac{\partial\mathcal{Q}}{\partial\tilde{\sigma}_{n,p}^{2,k}} = -\frac{1}{2}\sum_{t,j,m}\gamma_{jm}^k(t)\left\{\sum_{d}\frac{f_{jm,dp}^{2,k}\sigma_{n,p}^{2,k}}{\sigma_{jm,d}^{2,k}}\left(1 - \frac{(y_{t,d}^k - \mu_{jm,d}^k)^2}{\sigma_{jm,d}^{2,k}}\right)\right\} \tag{6.74}$$

$$\begin{aligned}\frac{\partial\mathcal{Q}^2}{\partial\tilde{\sigma}_{n,p}^{2,k}\partial\tilde{\sigma}_{nk,l}^2} = \frac{1}{2}\sum_{t,j,m}\gamma_{jm}^k(t)\Bigg\{&\sum_{d}\frac{f_{jm,dp}^{2,k}\sigma_{n,p}^{2,k}f_{jm,dl}^{2,k}\sigma_{n,l}^{2,k}}{\sigma_{jmk,d}^4}\left(1 - 2\frac{(y_{t,d}^k - \mu_{jm,d}^k)^2}{\sigma_{jm,d}^{2,k}}\right)\\ &- \delta(l-p)\sum_{d}\frac{f_{jm,dp}^{2,k}\sigma_{n,p}^{2,k}}{\sigma_{jm,d}^{2,k}}\left(1 - \frac{(y_{t,d}^k - \mu_{jm,d}^k)^2}{\sigma_{jm,d}^{2,k}}\right)\Bigg\}\end{aligned} \tag{6.75}$$

Therefore, the update formula can be given as:

$$\tilde{\boldsymbol{\Sigma}}_n^k = \tilde{\boldsymbol{\Sigma}}_{n,0}^k - \alpha\left[\left(\frac{\partial^2\mathcal{Q}}{\partial^2\tilde{\boldsymbol{\Sigma}}_n^k}\right)^{-1}\frac{\partial\mathcal{Q}}{\partial\tilde{\boldsymbol{\Sigma}}_n^k}\right]_{\tilde{\boldsymbol{\Sigma}}_n^k = \tilde{\boldsymbol{\Sigma}}_{n,0}^k} \tag{6.76}$$

where α is learning rate, which is usually set to be 1. In case of updating the variance of dynamic parameters, similar derivatives and update formulae can be obtained by simply adding Δ or Δ^2 to $y_{t,d}^k$,$\mu_{jm,d}^k$,$\sigma_{jm,d}^{2,k}$,$\sigma_{n,p}^{2,k}$, respectively. Note that it is important to make sure that the inequality in IEq-6.68 holds. If this inequality does not hold, the learning rate α may be reduced by a factor of 2.

6.2.2.2 Canonical Model Estimation

When performing canonical acoustic model estimation, noise model parameters are assumed to be fixed such that $\boldsymbol{\mu}_n^k = \boldsymbol{\mu}_{n,0}^k$ and $\boldsymbol{\mu}_h^k = \boldsymbol{\mu}_{h,0}^k$. Hence, the compensated cepstral mean becomes:

$$\boldsymbol{\mu}_{jm}^k \approx \boldsymbol{\mu}_{jm,0}^k + \mathbf{G}_{jm}^k(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) \tag{6.77}$$

Canonical Mean Update In order to obtain the optimal estimation of "pseudo-clean" model mean, the partial derivative with respect to $\boldsymbol{\mu}_{jm}$ is taken as:

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_{jm}} = \sum_{k,t} \gamma_{jm}^k(t) \left(\frac{\partial \boldsymbol{\mu}_{jm}^k}{\partial \boldsymbol{\mu}_{jm}}\bigg|_{\boldsymbol{\mu}_{jm,0}}^{T} \boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_t^k - \boldsymbol{\mu}_{jm}^k) \right) \tag{6.78}$$

$$= \sum_{k,t} \gamma_{jm}^k(t) \left(\mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \left(\mathbf{y}_t^k - \boldsymbol{\mu}_{jm,0}^k - \mathbf{G}_{jm}^k(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) \right) \right) \tag{6.79}$$

By letting this equation to zero, the update formula for model mean $\boldsymbol{\mu}_{jm}$ can be obtained as:

$$\boldsymbol{\mu}_{jm} = \boldsymbol{\mu}_{jm,0} + \left\{ \sum_{k,t} \gamma_{jm}^k(t) \mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \mathbf{G}_{jm}^k \right\}^{-1} \left\{ \sum_{k,t} \gamma_{jm}^k(t) \mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{jm}^{-1,k} (\mathbf{y}_t^k - \boldsymbol{\mu}_{jm,0}^k) \right\} \tag{6.80}$$

In case of dynamic parameters estimation, the expansion function is revised as:

$$\boldsymbol{\mu}_{\Delta jm}^k \approx \boldsymbol{\mu}_{\Delta jmk,0} + \mathbf{G}_{jm}^k(\boldsymbol{\mu}_{\Delta jm} - \boldsymbol{\mu}_{\Delta jm,0}) \tag{6.81}$$

Note that $\boldsymbol{\mu}_{\Delta h}$ and $\boldsymbol{\mu}_{\Delta n}$ are assumed to be zero in practice for convenience. Therefore, the update formulae for dynamic parameters can be similarly computed as:

$$\boldsymbol{\mu}_{\Delta jm} = \boldsymbol{\mu}_{\Delta jm,0} + \left\{ \sum_{k,t} \gamma_{jm}^k(t) \mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{\Delta jm}^{-1,k} \mathbf{G}_{jm}^k \right\}^{-1} \times \left\{ \sum_{k,t} \gamma_{jm}^k(t) \mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{\Delta jm}^{-1,k} (\Delta \mathbf{y}_t^k - \boldsymbol{\mu}_{\Delta jmk,0}) \right\} \tag{6.82}$$

and

$$\boldsymbol{\mu}_{\Delta^2 jm} = \boldsymbol{\mu}_{\Delta^2 jm,0} + \left\{ \sum_{k,t} \gamma_{jm}^k(t) \mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{\Delta^2 jm}^{-1,k} \mathbf{G}_{jm}^k \right\}^{-1} \times \left\{ \sum_{k,t} \gamma_{jm}^k(t) \mathbf{G}_{jm}^{T,k} \boldsymbol{\Sigma}_{\Delta^2 jm}^{-1,k} (\Delta^2 \mathbf{y}_t^k - \boldsymbol{\mu}_{\Delta^2 jm,0}^k) \right\} \tag{6.83}$$

Canonical Variance Update In case of updating the variance of D-dimension static parameters, the auxiliary function with respect to "pseudo-clean" acoustic model parameters can be re-written as:

$$\mathcal{Q} = \sum_{k,t} -\frac{1}{2}\gamma_{jm}^{k}(t)\left\{\sum_{d=1}^{D}\left(\log\sigma_{jm,d}^{2,k} + \frac{(y_{t,d}^{k}-\mu_{jm,d}^{k})^2}{\sigma_{jm,d}^{2,k}}\right)\right\} \tag{6.84}$$

In order to optimize the acoustic model's variance, Newton's method is adopted, which needs up to two order of derivatives with respect to $\tilde{\sigma}_{jm,p}^{2}$:

$$\frac{\partial\mathcal{Q}}{\partial\tilde{\sigma}_{jm,p}^{2}} = -\frac{1}{2}\sum_{k,t}\gamma_{jm}^{k}(t)\left\{\sum_{d}\frac{g_{jm,dp}^{2,k}\sigma_{jm,p}^{2}}{\sigma_{jm,d}^{2,k}}\left(1-\frac{(y_{t,d}^{k}-\mu_{jm,d}^{k})^2}{\sigma_{jm,d}^{2,k}}\right)\right\} \tag{6.85}$$

$$\begin{aligned}\frac{\partial\mathcal{Q}^2}{\partial\tilde{\sigma}_{jm,p}^{2}\partial\tilde{\sigma}_{jm,l}^{2}} = \frac{1}{2}\sum_{t,j,m}\gamma_{jm}^{k}(t)\Bigg\{&\sum_{d}\frac{g_{jm,dp}^{2,k}\sigma_{jm,p}^{2}g_{jm,dl}^{2,k}\sigma_{jm,l}^{2}}{\sigma_{jm,d}^{4,k}}\left(1-2\frac{(y_{t,d}^{k}-\mu_{jm,d}^{k})^2}{\sigma_{jm,d}^{2,k}}\right)\\ &-\delta(l-p)\sum_{d}\frac{g_{jm,dp}^{2,k}\sigma_{jm,p}^{2}}{\sigma_{jm,d}^{2,k}}\left(1-\frac{(y_{t,d}^{k}-\mu_{jm,d}^{k})^2}{\sigma_{jm,d}^{2,k}}\right)\Bigg\}\end{aligned} \tag{6.86}$$

Therefore, the update formula can be given as:

$$\tilde{\boldsymbol{\Sigma}}_{jm} = \tilde{\boldsymbol{\Sigma}}_{jm,0} - \alpha\left[\left(\frac{\partial^2\mathcal{Q}}{\partial^2\tilde{\boldsymbol{\Sigma}}_{jm}}\right)^{-1}\frac{\partial\mathcal{Q}}{\partial\tilde{\boldsymbol{\Sigma}}_{jm}}\right]_{\tilde{\boldsymbol{\Sigma}}_{jm}=\tilde{\boldsymbol{\Sigma}}_{jm,0}} \tag{6.87}$$

In case of updating the variance of dynamic parameters, similar derivatives and update formulae can be obtained by simply adding Δ or Δ^2 to $y_{t,d}^{k}$,$\mu_{jm,d}^{k}$,$\sigma_{jm,d}^{2,k}$,$\sigma_{jm,p}^{2}$, respectively.

6.2.3 Joint Adaptation and Adaptive Training

Different speaker has different speaking style such as female vs. male, native vs. non-native, young vs. elder, etc. Various rapid adaptation technologies have been proposed to improve the robustness against speaker variabilities. In this section, Maximum Likelihood Linear Regression MEAN (MLLR MEAN) [48] will be used to exploit the joint adaptation and adaptive training for speaker and noise. For convenience, noise compensation will be performed and followed by speaker adaptation for the joint adaptation. It is also assumed that one speaker may speak in multiple conditions and one condition only contains one speaker as one utterance represents one condition in the following formulation. Otherwise, MLLR estimation requires another first order Vector Taylor Series approximation [105] due to the nonlinear "g"-function in Eq-6.8. First, the "auxiliary" function of joint adaptive training is written as:

$$\mathcal{Q} = \sum_{s,k,t,j,m} \gamma_{jm}^{s,k}(t) \log p(\mathbf{y}_t^{s,k}|j,m,k,s) \tag{6.88}$$

$$\begin{aligned} &= \sum_{s,k,t,j,m} \gamma_{jm}^{s,k}(t) \left(-\frac{1}{2}\log|\boldsymbol{\Sigma}_{jm}^{k}| - \frac{1}{2}(\mathbf{y}_t^{k} - \boldsymbol{\mu}_{jm}^{s,k})^T \boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{jm}^{s,k}) \right) \\ &+ \sum_{s,k,t,j,m} \gamma_{jm}^{s,k}(t) \left(-\frac{1}{2}\log|\boldsymbol{\Sigma}_{\Delta jm}^{k}| - \frac{1}{2}(\Delta\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{\Delta jm}^{s,k})^T \boldsymbol{\Sigma}_{\Delta jm}^{-1,k}(\Delta\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{\Delta jm}^{s,k}) \right) \\ &+ \sum_{s,k,t,j,m} \gamma_{jm}^{s,k}(t) \left(-\frac{1}{2}\log|\boldsymbol{\Sigma}_{\Delta^2 jm}^{k}| - \frac{1}{2}(\Delta^2\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{\Delta^2 jm}^{s,k})^T \boldsymbol{\Sigma}_{\Delta^2 jm}^{-1,k}(\Delta^2\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{\Delta^2 jm}^{s,k}) \right) \end{aligned} \tag{6.89}$$

where s represents the speaker id, $\gamma_{jm}^{s,k}(t)$ is the component occupancy given the current adapted TVWR system using DNN based posterior features,

$$\gamma_{jm}^{s,k}(t) = \gamma_{j}^{s,k}(t) \frac{\tilde{c}_{jmt}(s,k,\hat{\boldsymbol{\Lambda}}) \mathcal{N}(\mathbf{y}_t^{s,k}; \hat{\boldsymbol{\mu}}_{jm}^{s,k}, \hat{\boldsymbol{\Sigma}}_{jm}^{k})}{\sum_{m=1}^{M} \tilde{c}_{jmt}(s,k,\hat{\boldsymbol{\Lambda}}) \mathcal{N}(\mathbf{y}_t^{s,k}; \hat{\boldsymbol{\mu}}_{jm}^{s,k}, \hat{\boldsymbol{\Sigma}}_{jm}^{k})} \tag{6.90}$$

$$= \gamma_{j}^{s,k}(t) \frac{\left(\sum_{i \in I_j} \hat{w}_{jmi} \tilde{p}(i|\boldsymbol{\chi}_t^{s,k}) \right) \hat{c}_{jm} \mathcal{N}(\mathbf{y}_t^{s,k}; \hat{\boldsymbol{\mu}}_{jm}^{s,k}, \hat{\boldsymbol{\Sigma}}_{jm}^{k})}{\sum_{m=1}^{M} \left(\sum_{i \in I_j} \hat{w}_{jmi} \tilde{p}(i|\boldsymbol{\chi}_t^{s,k}) \right) \hat{c}_{jm} \mathcal{N}(\mathbf{y}_t^{s,k}; \hat{\boldsymbol{\mu}}_{jm}^{s,k}, \hat{\boldsymbol{\Sigma}}_{jm}^{k})} \tag{6.91}$$

$\tilde{p}(i|\boldsymbol{\chi}_t^{s,k}) = p(i|\boldsymbol{\chi}_t^{s,k})/P(i)$ is the normalized posterior feature predicted by multi-style trained DNN, assuming that each speaker uses a global transform (it is easy to extend it to be regression class based adaptation), joint adapted acoustic models are given as (as noise compensation is performed separately for static and dynamic parameters, block MLLR mean transformation is assumed for convenience):

$$\boldsymbol{\mu}_{jm}^{s,k} \approx \mathbf{A}^s \boldsymbol{\mu}_{jm}^{k} + \mathbf{b}^s \tag{6.92}$$

$$\boldsymbol{\mu}_{\Delta jm}^{s,k} \approx \mathbf{A}_{\Delta}^s \boldsymbol{\mu}_{\Delta jm}^{k} + \mathbf{b}_{\Delta}^s \tag{6.93}$$

$$\boldsymbol{\mu}_{\Delta^2 jm}^{s,k} \approx \mathbf{A}_{\Delta^2}^s \boldsymbol{\mu}_{\Delta^2 jm}^{k} + \mathbf{b}_{\Delta^2}^s \tag{6.94}$$

where $\mathbf{A}^s$, $\mathbf{b}^s$ is the speaker dependent affine transform, VTS expansion functions are given as:

$$\boldsymbol{\mu}_{jm}^{k} \approx \boldsymbol{\mu}_{jm,0}^{k} + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{h}^{k} - \boldsymbol{\mu}_{h,0}^{k}) + \mathbf{F}_{jm}^{k}(\boldsymbol{\mu}_{n}^{k} - \boldsymbol{\mu}_{n,0}^{k}) \tag{6.95}$$

$$\boldsymbol{\mu}_{\Delta jm}^{k} \approx \boldsymbol{\mu}_{\Delta jm,0}^{k} + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{\Delta jm} - \boldsymbol{\mu}_{\Delta jm,0}) \tag{6.96}$$

$$\boldsymbol{\mu}_{\Delta^2 jm}^{k} \approx \boldsymbol{\mu}_{\Delta^2 jm,0}^{k} + \mathbf{G}_{jm}^{k}(\boldsymbol{\mu}_{\Delta^2 jm} - \boldsymbol{\mu}_{\Delta^2 jm,0}) \tag{6.97}$$

$$\boldsymbol{\Sigma}_{jm}^{k} \approx \mathbf{G}_{jm}^{k} \boldsymbol{\Sigma}_{jm} \mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^{k} \boldsymbol{\Sigma}_{n}^{k} \mathbf{F}_{jm}^{T,k} \tag{6.98}$$

$$\boldsymbol{\Sigma}_{\Delta jm}^{k} \approx \mathbf{G}_{jm}^{k} \boldsymbol{\Sigma}_{\Delta jm} \mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^{k} \boldsymbol{\Sigma}_{\Delta n}^{k} \mathbf{F}_{jm}^{T,k} \tag{6.99}$$

$$\boldsymbol{\Sigma}_{\Delta^2 jm}^{k} \approx \mathbf{G}_{jm}^{k} \boldsymbol{\Sigma}_{\Delta^2 jm} \mathbf{G}_{jm}^{T,k} + \mathbf{F}_{jm}^{k} \boldsymbol{\Sigma}_{\Delta^2 n}^{k} \mathbf{F}_{jm}^{T,k} \tag{6.100}$$

6.2.3.1 Speaker Transform Estimation

As the transform for static parameters shares similar objective function to that for dynamic parameters, only the speaker transform for static parameters are discussed with details. Given the auxiliary function to estimate the transform for a particular speaker:

$$\mathcal{Q} = \sum_{k,t,j,m} \gamma_{jm}^{s,k}(t) \left(-\frac{1}{2} \log |\boldsymbol{\Sigma}_{jm}^{k}| - \frac{1}{2} (\mathbf{y}_t^k - \boldsymbol{\mu}_{jm}^{s,k})^T \boldsymbol{\Sigma}_{jm}^{-1,k} (\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{jm}^{s,k}) \right) \tag{6.101}$$

Assuming that the speaker adaption transform is a global MLLR mean, both acoustic and noise models are given such that $\boldsymbol{\mu}_{jm}^{k} = \boldsymbol{\mu}_{jm,0}^{k}$, covariance matrices are diagonal, and defining that

$$\mathbf{W}^s = [\mathbf{A}^s \quad \mathbf{b}^s], \quad \boldsymbol{\zeta}_{jm}^{s,k} = [\boldsymbol{\mu}_{jm}^{T,s,k} \quad 1]^T \tag{6.102}$$

the auxiliary function can be re-written as:

$$\mathcal{Q} = K - \frac{1}{2} \sum_{d=1}^{D} \left(\mathbf{w}_d^s \mathbf{G}_d^s \mathbf{w}_d^{T,s} - 2 \mathbf{w}_d^s \mathbf{k}_d^{T,s} \right) \tag{6.103}$$

where $\mathbf{w}_d^s$ is the d^{th} row of $\mathbf{W}^s$, K is a independent constant term,

$$\mathbf{G}_d^s = \sum_{k,j,m} \sigma_{jm,d}^{-2,k} \boldsymbol{\zeta}_{jm}^{s,k} \boldsymbol{\zeta}_{jm}^{T,s,k} \sum_t \gamma_{jm}^{s,k}(t) \tag{6.104}$$

$$\mathbf{k}_d^s = \sum_{k,t,j,m} \gamma_{jm}^{s,k}(t) \sigma_{jm,d}^{-2,k} \mathbf{y}_{t,d}^k \boldsymbol{\zeta}_{jm}^{T,s,k} \tag{6.105}$$

Differentiating the auxiliary function with respect to $\mathbf{w}_d^s$ and equating to zero can yield the update formula:

$$\mathbf{w}_d^s = \mathbf{k}_d^s \mathbf{G}_d^{-1,s} \tag{6.106}$$

It is straightforward to extend it to be based on regression class by adjusting the sufficient statistics $\mathbf{G}_d^s$, $\mathbf{G}_d^s$ to be per regression class.

6.2.3.2 Noise Model Estimation

Assuming that the speaker transform and canonical model are given, the expansion function becomes:

$$\boldsymbol{\mu}_{jm}^{s,k} \approx \mathbf{A}^s \boldsymbol{\mu}_{jm,0}^k + \mathbf{A}^s \mathbf{G}_{jm}^k (\boldsymbol{\mu}_h^k - \boldsymbol{\mu}_{h,0}^k) + \mathbf{A}^s \mathbf{F}_{jm}^k (\boldsymbol{\mu}_n^k - \boldsymbol{\mu}_{n,0}^k) + \mathbf{b}^s \tag{6.107}$$

$$= \boldsymbol{\mu}_{jm,0}^{s,k} + \mathbf{G}_{jm}^{s,k} (\boldsymbol{\mu}_h - \boldsymbol{\mu}_{h,0}) + \mathbf{F}_{jm}^{s,k} (\boldsymbol{\mu}_n^k - \boldsymbol{\mu}_{n,0}^k) \tag{6.108}$$

where the joint Jacobian is written as:

$$\mathbf{G}_{jm}^{s,k} = \mathbf{A}^s \mathbf{G}_{jm}^k, \quad \mathbf{F}_{jm}^{s,k} = \mathbf{A}^s \mathbf{F}_{jm}^k, \quad \boldsymbol{\mu}_{jm,0}^{s,k} = \mathbf{A}^s \boldsymbol{\mu}_{jm,0}^k + \mathbf{b}^s \tag{6.109}$$

In order to estimate the noise mean parameters, derivatives are taken such as:

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_n^k} = \sum_{t,j,m} \gamma_{jm}^{s,k}(t) \left(\frac{\partial \boldsymbol{\mu}_{jm}^{s,k}}{\partial \boldsymbol{\mu}_n^k}\bigg|_{\boldsymbol{\mu}_{n,0}}^{T} \boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{jm}^{s,k}) \right) \tag{6.110}$$

$$= \sum_{t,j,m} \gamma_{jm}^{k}(t) \left(\mathbf{F}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \left(\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{jm,0}^{s,k} - \mathbf{G}_{jm}^{s,k}(\boldsymbol{\mu}_h^k - \boldsymbol{\mu}_{h,0}^k) - \mathbf{F}_{jm}^{s,k}(\boldsymbol{\mu}_n^k - \boldsymbol{\mu}_{n,0}^k) \right) \right) \tag{6.111}$$

and

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_h^k} = \sum_{t,j,m} \gamma_{jm}^{s,k}(t) \left(\frac{\partial \boldsymbol{\mu}_{jm}^{s,k}}{\partial \boldsymbol{\mu}_h^k}\bigg|_{\boldsymbol{\mu}_{n,0}}^{T} \boldsymbol{\Sigma}_{jm}^{-1,k}(\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{jm}^{s,k}) \right) \tag{6.112}$$

$$= \sum_{t,j,m} \gamma_{jm}^{k}(t) \left(\mathbf{G}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \left(\mathbf{y}_t^{s,k} - \boldsymbol{\mu}_{jm,0}^{s,k} - \mathbf{G}_{jm}^{s,k}(\boldsymbol{\mu}_h^k - \boldsymbol{\mu}_{h,0}^k) - \mathbf{F}_{jm}^{s,k}(\boldsymbol{\mu}_n^k - \boldsymbol{\mu}_{n,0}^k) \right) \right) \tag{6.113}$$

After some algebra manipulations and setting above derivatives to zero, an equation system can be obtained as:

$$\mathbf{A}_s \boldsymbol{\mu}_h^k + \mathbf{B}_s \boldsymbol{\mu}_n^k = \mathbf{g}_s \tag{6.114}$$

$$\mathbf{D}_s \boldsymbol{\mu}_h^k + \mathbf{K}_s \boldsymbol{\mu}_n^k = \mathbf{h}_s \tag{6.115}$$

where

$$\mathbf{A}_s = \sum_{t,j,m} \gamma_{jm}^{k}(t) \mathbf{F}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \mathbf{G}_{jm}^{s,k} \tag{6.116}$$

$$\mathbf{B}_s = \sum_{t,j,m} \gamma_{jm}^{k}(t) \mathbf{F}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \mathbf{F}_{jm}^{s,k} \tag{6.117}$$

$$\mathbf{D}_s = \sum_{t,j,m} \gamma_{jm}^{k}(t) \mathbf{G}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \mathbf{G}_{jm}^{s,k} \tag{6.118}$$

$$\mathbf{K}_s = \sum_{t,j,m} \gamma_{jm}^{k}(t) \mathbf{G}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} \mathbf{F}_{jm}^{s,k} \tag{6.119}$$

$$\mathbf{g}_s = \sum_{t,j,m} \gamma_{jm}^{k}(t) \mathbf{F}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} (\mathbf{y}_t^{k} - \boldsymbol{\mu}_{jm,0}^{s,k} + \mathbf{G}_{jm}^{s,k} \boldsymbol{\mu}_{h,0}^{k} + \mathbf{F}_{jm}^{s,k} \boldsymbol{\mu}_{n,0}^{k}) \tag{6.120}$$

$$\mathbf{h}_s = \sum_{t,j,m} \gamma_{jm}^{s,k}(t) \mathbf{G}_{jm}^{T,s,k} \boldsymbol{\Sigma}_{jm}^{-1,k} (\mathbf{y}_t^{k} - \boldsymbol{\mu}_{jm,0}^{s,k} + \mathbf{G}_{jm}^{k} \boldsymbol{\mu}_{h,0}^{k} + \mathbf{F}_{jm}^{s,k} \boldsymbol{\mu}_{n,0}^{k}) \tag{6.121}$$

Note that $\mathbf{A}_s = \mathbf{K}_s^T$. Solving above linear system can give the following closed form update formulae used for parameter estimation:

$$\boldsymbol{\mu}_h^k = (\mathbf{A}_s - \mathbf{B}_s \mathbf{K}_s^{-1} \mathbf{D}_s)^{-1} (\mathbf{g}_s - \mathbf{B}_s \mathbf{K}_s^{-1} \mathbf{h}_s) \tag{6.122}$$

$$\boldsymbol{\mu}_n^k = (\mathbf{B}_s - \mathbf{A}_s \mathbf{D}_s^{-1} \mathbf{K}_s)^{-1} (\mathbf{g}_s - \mathbf{A}_s \mathbf{D}_s^{-1} \mathbf{h}_s) \tag{6.123}$$

$$= (\mathbf{K}_s - \mathbf{D}_s \mathbf{A}_s^{-1} \mathbf{B}_s)^{-1} (\mathbf{h}_s - \mathbf{D}_s \mathbf{A}_s^{-1} \mathbf{g}_s) \tag{6.124}$$

As variance parameters are not adapted to speaker, the derivatives for noise variance update are almost the same as NAT in Eq-6.74 and Eq-6.75, except that $\mu^{k}_{jm,d}$ is replaced by $\mu^{s,k}_{jm,d}$.

6.2.3.3 Canonical Model Estimation

As mentioned earlier, joint adaptation is assumed to be performed for noise followed by speaker such as

$$\boldsymbol{\mu}^{s,k}_{jm} \approx \mathbf{A}^{s}\boldsymbol{\mu}^{k}_{jm,0} + \mathbf{A}^{s}\mathbf{G}^{k}_{jm}(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) + \mathbf{b}^{s} \tag{6.125}$$

$$= \boldsymbol{\mu}^{s,k}_{jm,0} + \mathbf{G}^{s,k}_{jm}(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) \tag{6.126}$$

In order to obtain the optimal estimation of "pseudo-clean" model mean, the partial derivative with respect to $\boldsymbol{\mu}_{jm}$ is taken as:

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_{jm}} = \sum_{s,k,t} \gamma^{s,k}_{jm}(t) \left(\frac{\partial \boldsymbol{\mu}^{s,k}_{jm}}{\partial \boldsymbol{\mu}_{jm}}\bigg|^{T}_{\boldsymbol{\mu}_{jm,0}} \boldsymbol{\Sigma}^{-1,k}_{jm}(\mathbf{y}^{s,k}_{t} - \boldsymbol{\mu}^{s,k}_{jm}) \right) \tag{6.127}$$

$$= \sum_{s,k,t} \gamma^{s,k}_{jm}(t) \left(\mathbf{G}^{T,s,k}_{jm}\boldsymbol{\Sigma}^{-1,k}_{jm} \left(\mathbf{y}^{k}_{t} - \boldsymbol{\mu}^{s,k}_{jm,0} - \mathbf{G}^{s,k}_{jm}(\boldsymbol{\mu}_{jm} - \boldsymbol{\mu}_{jm,0}) \right) \right) \tag{6.128}$$

By letting this equation to zero, the update formula for model mean $\boldsymbol{\mu}_{jm}$ can be obtained as:

$$\boldsymbol{\mu}_{jm} = \boldsymbol{\mu}_{jm,0} + \left\{ \sum_{s,k,t} \gamma^{s,k}_{jm}(t) \mathbf{G}^{T,s,k}_{jm}\boldsymbol{\Sigma}^{-1,k}_{jm}\mathbf{G}^{s,k}_{jm} \right\}^{-1} \left\{ \sum_{s,k,t} \gamma^{s,k}_{jm}(t) \mathbf{G}^{T,s,k}_{jm}\boldsymbol{\Sigma}^{-1,k}_{jm}(\mathbf{y}^{s,k}_{t} - \boldsymbol{\mu}^{s,k}_{jm,0}) \right\} \tag{6.129}$$

Therefore, the update formulae for dynamic parameters can be similarly computed as:

$$\boldsymbol{\mu}_{\Delta jm} = \boldsymbol{\mu}_{\Delta jm,0} + \left\{ \sum_{s,k,t} \gamma^{s,k}_{jm}(t) \mathbf{G}^{T,s,k}_{\Delta jm}\boldsymbol{\Sigma}^{-1,k}_{\Delta jm}\mathbf{G}^{s,k}_{\Delta jm} \right\}^{-1} \times \left\{ \sum_{s,k,t} \gamma^{s,k}_{jm}(t) \mathbf{G}^{T,s,k}_{\Delta jm}\boldsymbol{\Sigma}^{-1,k}_{\Delta jm}(\Delta\mathbf{y}^{s,k}_{t} - \boldsymbol{\mu}^{s,k}_{\Delta jm,0}) \right\} \tag{6.130}$$

where the dynamic joint Jacobian is given as:

$$\mathbf{G}^{s,k}_{\Delta jm} = \mathbf{A}^{s}_{\Delta}\mathbf{G}^{k}_{jm}, \quad \boldsymbol{\mu}^{s,k}_{jm,0} = \mathbf{A}^{s}_{\Delta}\boldsymbol{\mu}^{k}_{\Delta jm,0} + \mathbf{b}^{s}_{\Delta} \tag{6.131}$$

As no speaker adaptation is performed on the variance parameters, the derivatives for canonical variance update are almost the same as NAT in Eq-6.85 and Eq-6.86, except that $\mu^{k}_{jm,d}$ is replaced by $\mu^{s,k}_{jm,d}$.

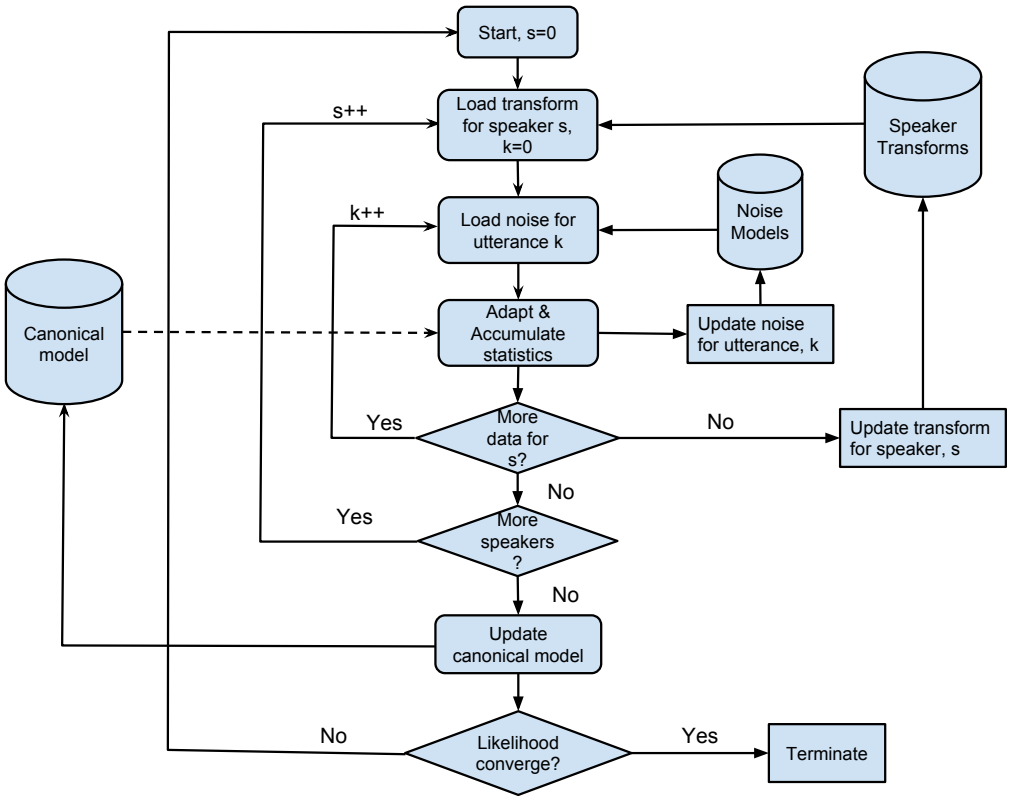

Figure 6.1: *A diagram of joint adaptive training for TVWR.*

6.2.3.4 Training Algorithm

In this section, joint adaptive training algorithm is summarized in the Figure. 6.1:

1. Initialize the canonical model using speaker/noise independent model, use the head/tail 20 frames to initialize the noise model with zero channel mean, initialize the speaker transform (MLLR mean) using identity matrix and zero bias, initialize the speaker index as $s = 0$.

2. Load speaker transform (MLLR mean) for speaker s.

3. Load noise model for utterance k by speaker s.

4. Load the current canonical model to perform VTS noise compensation and then (MLLR mean) speaker adaptation.

5. Perform forward-backward estimation, and accumulate sufficient statistics for noise, speaker transform and canonical acoustic model estimation, respectively. Then, update noise model and store it into the disk.

6. If more utterances by speaker s are available, increase k by 1 and then goto step-3; otherwise, update speaker transform and store it into the disk and goto step-7.

7. If there exist more speakers, increase s by 1 and then goto step-2; otherwise, update canonical model and store it into the disk and goto step-8.

8. If the training likelihood converges, then terminate; otherwise, set $s = 0$ and goto step-2.

Once the training is done, speaker transforms and noise models can be discarded during recognition. In order to apply canonical model for recognition, speaker independent model will be used to decode and generate hypotheses, which serves as the unsupervised transcription to estimate the initial speakers transform and noise models. The canonical model adapted by these initial speaker transform and noise models can generate the new hypotheses, which will be used to update speaker transform and noise models. Iterative recognition and estimation may be applied till the recognition converges.

6.2.4 Experimental Results

In this section, experiments were conducted on the Aurora4 corpus for evaluation of robust TVWR using DNN based posteriors. Two training sets, clean and multi-style, are available and 16kHz data were used in this experiment. Each training dataset comprises 7138 utterances from 84 speakers, or about 12 hours of speech. In the clean training dataset, all data were recorded by a close-talking microphone. Half of multi-style training data came from the same microphone, but the rest were recorded by a variety of different microphones. The multi-style training data contains 6 different types of noise. The noise was added at a randomly chosen SNR between 10 and 20 dB, averaged 15 dB. For evaluation, 14 test sets (each with 330 utterances and 8 speakers) are available for covering 14 test conditions, which can be further grouped into 4 sets, named as A, B, C, D. A represents the clean test set; B contains 6 test sets and 6 types of noise, whose SNR is randomly chosen between 5 and 15 dB, averaged 10 dB; C is also the clean test set but recorded with a different microphone; D is similar to B, except that the recording devices are different.

The baseline GMM systems consisted of context-dependent HMMs with 3187 senones (or tied states) and 16 mixtures per state using maximum likelihood estimation. The input features for noise adaptation and adaptive training are Mel Frequency Cepstral Coefficients (MFCC), including 12 static parameters, zero-th coefficient, and first two orders of dynamic parameters. Cepstral mean&variance normalization (CMVN) was also used to perform two-model re-estimation for another baseline. The GMM baseline using CMVN features was used to align the training data as the initial senone labels for DNN training. The testing results were obtained by a bigram full decoding using HTK.

The input features for DNN training are 24-dimensional log Mel filter-bank (FBANK) features with two order dynamic parameters and context expansion with window size 11, i.e. 792 visible units. There were 6 hidden layers with sigmoid activation function and 2048 hidden units for each layer. Output layer had 3187 units corresponding to the context-dependent senones and used soft-max as activation function. Layer-by-layer generative pre-training was performed to obtain an initialization of DNN. Cross-entropy criterion was used for back propagation fine tuning. Back propagation was down using stochastic gradient descent with 128 examples per mini-batch. The initial learning rate was 0.015, which was reduced by a factor of 2 if cross validation performance was dropped. Momentum was 0 and maximum training iteration was 25. 1206 multi-style development utterances were extracted from the corpus. The trained DNN was then used to re-align the training data, and the second DNN was trained with updated labels, which was the final DNN for the following experiments. The DNN output log posterior features were analyzed by Principle Component Analysis (PCA), and 13 dimensional features were projected and appended to the original CMVN features. The result features were used to train a tandem system.

One fMLLR [131] transform was used per speaker to train a GMM system, which was also used for speaker adaptive training of another DNN system. Due to the limitation of current formulation and implementation, only MLLR mean adaptation was used for speaker adaptation and adaptive training in the TVWR framework. To perform unsupervised speaker or noise adaptation, one full bigram decoding was performed first and followed by an EM estimation of noise model or speaker transform. 0.001 was used as the floor of building the confusion table for sparse regression of TVWR.

First, various baseline systems trained by multi-style data are reported in Table. 6.3 (compact version) and Table. 6.5 (full version). GMM baseline using MFCC performed worst among all the systems. This tells that the conventional acoustic features are not robust against noisy environments. Cepstral mean&variance normalization (CMVN) significantly improved the recognition results on all test sets. It is also comparable to Advanced feature enhancement (AFE) [105]. Hence, the initial DNN training label was obtained by aligning CMVN features. DNN using filter-bank features achieved dramatical recognition error reduction and performed best among all. Its performance is also comparable to the report [127]. This DNN was then used to generate posterior features to train all following TVWR and tandem systems. Tandem system significantly outperformed other GMM based systems due to the additional information from DNN. As dimension reduction was performed to build the tandem features, information loss makes it perform worse than the DNN baseline. Different from Tandem system, TVWR uses the raw posterior features but performs sparse regression. Finally, TVWR achieved better performance than other GMM and combination systems. However, TVWR performed worse than DNN baseline. Considering the poor performance of GMM baseline system using MFCC features, one possible explanation is that GMM part of TVWR was giving incorrect likelihood and

misleading the decision. In order to obtain better performance from TVWR, GMM system should be more complementary to DNN system. One way to improve the TVWR performance is to boost the performance of GMM part, such as discriminative training, speaker and noise adaptation.

System		A	B	C	D	Avg
Model	Feature					
GMM	MFCC	10.5	23.6	29.4	38.1	27.7
GMM	CMVN	8.0	16.2	18.1	32.2	22.6
GMM [105]	AFE	8.8	16.7	19.1	28.6	21.4
DNN	FBANK	**4.8**	**8.8**	**8.8**	**20.5**	**13.6**
Tandem	CMVN+PCA	6.4	12.4	13.0	28.0	18.7
TVWR	MFCC	5.6	9.1	11.3	21.7	14.4
DNN [127]	FBANK	5.6	8.8	8.9	20.0	13.4

Table 6.3: *Compact recognition results (WER%) of various baseline systems without adaptation on Aurora4.*

Second, various system performances based on adaptation and adaptive training are shown in Table. 6.4 (compact version) and Table. 6.6 (full version). In order to apply VTS noise compensation, acoustic features used in this experiment were MFCC. If no DNN information was used, multi-style trained GMM can greatly benefit from adaptation techniques and achieved significant improvements over the multi-style GMM baseline. In addition, adaptive training also helped to improve the multi-style trained system performance as speaker and noise variations can be removed during training so that a better canonical model can be adapted for better recognition accuracy. With both speaker (MLLR mean) and noise adaptations, GMM [105] obtained comparable performance to DNN using FBANK. Adaptation and adaptive training also helped to improve Tandem system performance, however, it is much worse than the DNN baseline system without any adaptation. As explained earlier, TVWR uses full dimensional posterior features but performs sparse regression, TVWR obtained better performance than Tandem system without any adaptation. On the other hand, GMM parameters of TVWR were estimated based on MFCC features, which is the assumption of VTS adaptation for noise compensation. After either VTS or MLLR mean adaptation, TVWR obtained 1.8% absolute improvements over the DNN baseline system. This tells that the GMM adaptation is really powerful and provides rich complementary information to the DNN. If joint adaptation was performed, another 0.6% absolute improvements were obtained. After speaker adaptive training (SAT), TVWR with speaker adaptation achieved 2.4% absolute gain over DNN baseline and 1.2% absolute gain over speaker adaptive trained DNN. Joint

adaptation further pushed the system performance to 10.9% WER. In the TVWR framework, noise adaptive training seems not to help much compared to the simple multi-style training in average. However, NAT helped to obtain better results on the test set A (clean) and C (channel distortion), which should be expected as NAT is supposed to be able to remove the noise variation. After joint adaptation and adaptive training, best recognition results were observed among all TVWR systems.

Finally, when compared to the results on Aurora4 from literatures [127, 132, 133, 134], all TVWR systems after either VTS and MLLR mean adaptation outperformed the best system using multi-style training data from literatures. If only considering the noise factor, TVWR still achieved 0.6% absolute improvements over NAT+Dropout based DNN system. When only the speaker factor is considered, TVWR+SAT obtained 1.2% absolute improvements over the DNN+SAT system. So far, all TVWR systems were based on the posterior features from the baseline DNN system, i.e. 13.6% WER. In other words, with the help of adaptation and adaptive training in the TVWR framework, 2.9% absolute (or 20% relative) improvements can be obtained. It is interesting to note that TVWR is particular helpful for channel distortion, which leads to much larger relative improvements over the other systems on the test sets C(channel distortion) and D (channel distortion + additive noise). This largely benefits from the good environmental model and good approximation via VTS [58], which however have not been studied well for the DNN systems. It is also important to note that the best literature result [134] needs the stereo training data (clean training data + multi-style training data) and 4 DNNs (for masking, feature mapping, posterior average). The second best literature [133] also needs stereo training data and 3 DNNs (for masking, posterior average). Our system, on the other hand, only used multi-style training data and one un-adapted DNN. These results tell that combination and adaptation via TVWR are really powerful for robust speech recognition.

6.2.5 Summary

In this section, joint speaker/noise adaptive training and adaptation has been proposed for TVWR, a framework that combines the discriminative power of the DNN and the adaptability of the GMM for robust automatic speech recognition. On top of the superior performance of the DNN-based system, we were able to obtain further improvements on the Aurora 4 test sets by applying either the speaker adaptive training or noise adaptive training to the GMM parameters of the TVWR systems. Finally, the best overall word error rate performance of 10.7% was obtained by applying joint adaptive training, which was 21.3% relatively better than the DNN baseline system and also better than the best report in the current literatures.

System		Adaptation		A	B	C	D	Avg
Model	Training	Noise	Speaker					
GMM	multi-style	VTS	-	9.7	14.9	14.0	22.4	17.7
		-	MLLR	7.5	15.2	12.2	26.0	19.1
		VTS	MLLR	7.3	12.7	10.7	20.4	15.5
	SAT	-	MLLR	7.0	13.2	9.7	22.3	16.4
		VTS	MLLR	6.5	11.4	9.8	19.2	14.3
	NAT	VTS	-	8.5	13.7	11.4	21.3	16.4
		VTS	MLLR	6.6	11.9	8.9	19.5	14.6
	Joint	VTS	MLLR	6.3	11.2	8.1	18.4	13.7
	Joint [105]	VTS	MLLR	5.6	11.0	8.8	17.8	13.4
Tandem	multi-style	-	fMLLR	5.9	11.5	10.9	26.3	17.4
	SAT	-	fMLLR	5.6	11.0	10.0	25.6	16.8
TVWR	multi-style	VTS	-	5.6	8.3	8.3	16.8	11.8
		-	MLLR	5.0	8.1	7.3	17.4	11.8
		VTS	MLLR	5.0	7.8	7.7	16.1	11.2
	SAT	-	MLLR	5.0	7.8	**6.7**	16.5	11.2
		VTS	MLLR	5.1	7.6	7.0	15.9	10.9
	NAT	VTS	-	4.9	8.5	7.7	17.1	11.9
		VTS	MLLR	4.7	8.0	7.3	16.1	11.2
	Joint	VTS	MLLR	4.4	7.5	7.1	**15.6**	**10.7**
DNN	NAT [127]	VTS	-	5.4	8.8	7.8	19.6	13.1
	Dropout [127]	-	-	5.1	8.4	8.6	19.3	12.9
	NAT+Dropout [127]	VTS	-	5.4	8.3	7.6	18.5	12.4
	stereo [132]	cFDLR	-	5.1	8.5	8.4	17.6	12.1
	stereo [133]	LIN	-	4.6	7.5	7.6	17.0	11.4
	stereo [134]	masking	-	4.5	**7.4**	8.1	16.5	11.1
	SAT	-	fMLLR	**4.3**	7.9	6.7	19.2	12.4

Table 6.4: *Compact recognition results (WER%) of various systems on Aurora4 based on adaptation and adaptive training.*

System		A	B							C	D					Avg
Model	Feature	1	2	3	4	5	6	7	Avg	8	9	10	11	12.00	13.00	14.00	Avg	
GMM	MFCC	10.5	14.0	19.1	21.9	23.6	19.7	16.2	23.6	29.4	31.8	37.1	38.8	43.3	35.5	42.4	38.1	27.7
GMM	CMVN	8.0	11.3	15.6	17.8	17.4	14.4	20.7	16.2	18.1	22.6	32.9	33.2	35.5	31.5	32.2	37.5	22.6
GMM [105]	AFE	8.8	13.1	15.9	20.0	18.4	15.1	17.4	16.7	19.1	24.1	27.8	31.1	30.9	28.6	29.4	28.6	21.4
DNN	FBANK	**4.8**	**6.1**	**8.6**	10.8	**9.1**	**8.1**	**10.2**	**8.8**	**8.8**	**13.0**	**21.8**	**24.3**	**21.3**	21.1	**21.7**	**20.5**	**13.6**
Tandem	CMVN+PCA	6.4	7.7	12.4	15.0	13.4	12.0	14.1	12.4	13.0	18.9	29.5	30.5	30.4	28.2	30.5	28.0	18.7
TVWR	MFCC	5.6	6.3	8.7	**10.1**	10.3	8.4	10.6	9.1	11.3	14.8	22.0	24.3	23.9	**20.6**	24.2	21.7	14.4

Table 6.5: *Full recognition results (WER%) of various baseline systems without adaptation on Aurora4.*

System		Adaptation		A	B							C	D							Avg
Model	Training	Noise	Speaker	1	2	3	4	5	6	7	Avg	8	9	10	11	12	13	14	Avg	
GMM	multi-style	VTS	-	9.7	10.9	14.9	16.4	15.8	14.3	16.9	14.9	14.0	15.2	22.5	24.0	24.0	21.9	26.8	22.4	17.7
		-	MLLR	7.5	8.4	14.6	18.1	18.7	12.6	19.1	15.2	12.2	15.9	26.3	29.0	30.0	23.6	31.5	26.0	19.1
		VTS	MLLR	7.3	8.6	12.1	15.0	13.7	11.8	14.9	12.7	10.7	12.1	21.2	22.3	22.5	19.8	24.7	20.4	15.5
	SAT	-	MLLR	7.0	7.7	13.0	15.2	15.3	11.2	16.8	13.2	9.7	12.9	22.4	25.2	25.5	20.2	27.9	22.3	16.4
		VTS	MLLR	6.5	7.1	11.2	13.0	12.8	10.1	14.4	11.4	9.8	10.9	20.1	21.3	21.7	17.4	23.8	19.2	14.3
	NAT	VTS	-	8.5	10.1	13.4	15.3	14.3	12.9	16.0	13.7	11.4	12.9	21.7	23.3	23.2	20.7	25.9	21.3	16.4
		VTS	MLLR	6.6	8.0	12.0	13.9	12.8	10.9	14.1	11.9	8.9	11.4	20.3	21.7	21.7	18.3	23.6	19.5	14.6
	Joint	VTS	MLLR	6.3	7.2	11.2	13.2	11.9	10.1	13.5	11.2	8.1	11.0	19.3	20.7	20.2	17.0	22.5	18.4	13.7
	Joint [105]	VTS	MLLR	5.6	6.7	10.5	13.4	12.1	9.5	13.8	11.0	8.8	10.3	17.8	20.7	20.8	16.5	20.8	17.8	13.4
Tandem	multi-style	-	fMLLR	5.9	6.7	11.5	14.7	12.4	10.5	13.1	11.5	10.9	16.7	28.0	29.6	28.1	26.9	28.4	26.3	17.4
	SAT	-	fMLLR	5.6	6.3	11.0	14.0	11.8	9.9	13.0	11.0	10.0	15.2	27.1	29.0	27.7	26.6	28.0	25.6	16.8
TVWR	multi-style	VTS	-	5.6	5.6	7.9	10.1	9.0	7.7	9.7	8.3	8.3	10.0	17.8	20.0	17.9	16.4	18.9	16.8	11.8
		-	MLLR	5.0	5.2	7.8	9.3	9.4	7.4	9.8	8.1	7.3	9.7	18.0	20.2	19.5	16.9	19.9	17.4	11.8
		VTS	MLLR	5.0	5.4	7.5	9.4	8.4	7.2	9.2	7.8	7.7	9.0	16.9	19.4	17.1	16.0	18.4	16.1	11.2
	SAT	-	MLLR	5.0	5.2	7.2	9.0	9.0	7.2	9.5	7.8	**6.7**	9.1	17.1	19.6	17.8	16.1	19.0	16.5	11.2
		VTS	MLLR	5.1	5.3	**7.1**	**8.8**	8.6	7.0	8.9	7.6	7.0	**8.9**	16.5	**18.9**	17.2	**15.1**	18.5	15.9	10.9
	NAT	VTS	-	4.9	5.9	8.2	9.9	9.3	7.9	10.0	8.5	7.7	10.3	17.7	20.1	18.3	16.9	19.1	17.1	11.9
		VTS	MLLR	4.7	5.4	7.6	9.5	8.9	7.3	9.3	8.0	7.3	9.1	16.6	19.2	17.5	16.0	18.1	16.1	11.2
	Joint	VTS	MLLR	4.4	5.0	**7.1**	9.2	**8.1**	**6.7**	8.9	7.5	7.1	9.0	**16.3**	19.0	**16.6**	15.3	**17.5**	**15.6**	**10.7**
DNN	NAT [127]	VTS	-	5.4	-	-	-	-	-	-	8.8	7.8	-	-	-	-	-	-	19.6	13.1
	Dropout [127]	-	-	5.1	-	-	-	-	-	-	8.4	8.6	-	-	-	-	-	-	19.3	12.9
	NAT+Dropout [127]	VTS	-	5.4	-	-	-	-	-	-	8.3	7.6	-	-	-	-	-	-	18.5	12.4
	stereo [132]	cFDLR	-	5.1	-	-	-	-	-	-	8.5	8.4	-	-	-	-	-	-	17.6	12.1
	stereo [133]	LIN	-	4.6	**4.9**	7.6	9.0	8.4	7.3	**7.9**	7.5	7.6	9.4	18.3	20.5	18.7	16.3	18.8	17.0	11.4
	stereo [134]	masking	-	4.5	-	-	-	-	-	-	**7.4**	8.1	-	-	-	-	-	-	16.5	11.1
	SAT	-	fMLLR	**4.3**	5.0	7.8	9.7	8.5	7.2	9.0	7.9	6.7	10.1	21.0	23.0	20.7	18.8	21.7	19.2	12.4

Table 6.6: *Complete recognition results (WER%) of various systems on Aurora4 based on adaptation and adaptive training.*

Chapter 7

Conclusions and Future Works

7.1 Conclusions

In this thesis, an implicit trajectory model using Temporally Varying Weight Regression (TVWR) has been proposed to learn the importance of Gaussian components under different acoustic contexts. This allows the temporal varying attributes of speech to be better recognized. Instead of modelling the temporal correlations directly using a long span of acoustic features, contextual information is implicitly incorporated into TVWR using posterior features. As a result, TVWR is able to model non-stationary GMM distributions whose temporally varying Gaussian component weights are obtained through regression with the posterior features. Although TVWR is similar to fMPE in terms of modeling time-varying model parameters using the posteriors derived from long span acoustic features, the underlying formulation and model parameterization are different. Specifically, TVWR follows a proper probabilistic formulation that yields a much simpler parameter estimation compared to the fMPE. Moreover, both maximum likelihood and discriminative training parameter estimation formulae can be derived. Experiments were conducted on the Wall Street Journal (CSR-WSJ0+WSJ1) corpora for 20k open vocabulary continuous speech recognition and the Aurora 4 corpus for the 5k closed vocabulary noisy speech recognition. Experimental results show that TVWR achieves better performance compared to the standard HMM system for both maximum likelihood and minimum phone error training. Moreover, the discriminatively trained TVWR models also achieved comparable (or marginally better) performance compared to fMPE.

After that, TVWR has been studied for cross-lingual speech recognition. Particularly, multi-stream TVWR has been proposed to boost the recognition accuracy. Richer context information can be obtained by introducing temporal and spatial context expansion. Although more computing costs will be introduced, this may not be an issue if the recognition performance is more important. This approach has been evaluated on the task of building ASR with limited resources. Experiments show that multi-stream TVWR with

limited training data (1h English or 6h Malay) achieved the performance of only less than 2% inferior to the respective fullset English (15h) and Malay (74h) baseline systems.

Next, TVWR has been investigated as an approach of combining GMM and DNN to achieve a high quality and adaptable state probability model for automatic speech recognition. The resulting GMM+DNN/HMM system is different from the tandem systems in that the GMMs are trained directly on the Cepstral acoustic features, rather than the DNN-derived tandem features. Posterior grouping and sparse regression have been developed to address the issue of incorporating the high dimensional CD-DNN posteriors into TVWR without dramatically increasing the system complexity. Experiments shows that the proposed system has significantly outperformed the DNN baseline system and the speaker adapted DNN system.

Finally, adaptation and adaptive training of TVWR has been formulated for robust speech recognition. When the posteriors for the weight regression are generated by a GMM, noise model compensation can be applied to both the GMM front-end as well as the GMM parameters of TVWR. TVWR has also been investigated as an approximation of noise adaptive training. Experiments have shown that TVWR has obtained significantly better performance than the conventional GMM systems after adaptation. On the other hand, when the posteriors are generated by the DNN, speaker, noise, joint adaptation and adaptive training have been developed for the TVWR system. In order to achieve better performance, the DNN is trained using the multi-style training data. Experiments on the Aurora 4 corpus have shown that the joint speaker and noise adapted TVWR system (10.7% WER) outperformed the DNN baseline system (13.6% WER) by 21.3% (relative) and also outperformed the best result (11.1% WER) [134] in the current literatures.

7.2 Future Works

In this section, several possible future works beyond the scope of this thesis are suggested, which aim to improve the current system or incorporate with other techniques.

- Currently, TVWR for building the ASR system with limited resources only uses the context-independent posteriors from the foreign shallow neural networks. One possible further work is to use better context-dependent posteriors from the foreign deep neural networks. Due to the superiority of the NN posteriors, (temporal and spatial) context expansion and a separate state clustering can be used to implicitly increase the importance of the posterior information. More flexible and explicit weighting the contribution of the NN posteriors needs to be investigated. For example, a global factor may be added to the log regressed value and tuned according to a development set.

- Although subspace GMM [135] has been less popular due to the introduction of the DNN, subspace GMM has more compact model parameterization and performs better than the conventional GMM in general tasks. Subspace GMM uses sharable projections and component specific vectors to estimate the statistics of the Gaussian components, and uses soft-max regression function to obtain the static weights. In other words, subspace GMM still has the basic GMM structure, therefore, it is possible to add a time-varying factor to the static weight. A better GMM variant may be more complementary to the DNN systems such that better performance may be expected.

- So far, adaptation and adaptive training of TVWR are only formulated using Maximum Likelihood criterion, while discriminative training has been widely used to build the state-of-the-art GMM-based system due to its superiority. This training criterion can be used to perform speaker adaptive training [136, 137] and noise adaptive training [138]. These techniques may also be applied to the TVWR framework to obtain a more robust system. On the other hand, only a global MLLR mean transform was used for joint speaker and noise adaptation in this thesis. It is possible to use more transforms by regression classes and more powerful constrained MLLR (CMLLR) [46]transforms.

- The deep learning algorithm has been applied to other neural network variants, such as convolutional neural network (CNN) [139, 140, 141] and recurrent neural network (RNN) [142]. In addition, dropout training [143], feature enhancement [127, 129, 132], sequential training [144, 145] and many other techniques have been developed for improving the deep neural networks. It is interesting to explore combining other neural networks and the GMM in the TVWR framework for robust speech recognition. For example, this investigation can be used to verify that whether the adapted GMM can be still complementary to the CNN or the CNN has the ability to remove more speech variabilities.

- In the current TVWR framework, the posterior generator is assumed to be as an independent external component. This assumption makes the framework is more flexible to use any probabilistic classifier. However, recent progress suggests that DNN is a superior acoustic model for speech recognition. It is interesting to embed the DNN training into the TVWR framework so that GMM and DNN can better contribute to each other during training. For example, if maximum likelihood estimation is performed, the objective function for DNN training may become:

$$\mathcal{Q}_{ML} = \sum_{t,j,m,i} \gamma_{jmi}^{ML}(t) \left\{ \log p(i|\boldsymbol{\tau}_t) - \log p(i) \right\} \tag{7.1}$$

, which is very similar to the cross-entropy function used for conventional DNN training:

$$\mathcal{F}_{CE} = -\sum_{t}\sum_{i}\delta(i - i_t)\log p(i|\boldsymbol{\tau}_t) \tag{7.2}$$

where $\delta()$ is a Dirac delta function, i_t is the label of latent variable at time t. The main difference between these two functions are that the conventional DNN training uses hard labels while embedded training uses soft labels. One advantage of embedded training is that there is need not to know what the latent variable really is but to define the number of latent variables to focus on the training objective. Similarly, embedded training may also be applied with discriminative training criteria [144, 145].

References

[1] C.M. Bishop and N.M. Nasrabadi, *Pattern recognition and machine learning*, vol. 1, springer New York, 2006. 1, 4, 13, 14, 45, 105

[2] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, feb 1989. 1, 8, 52, 57

[3] B.H. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 307–309, 1986. 1, 8, 29, 52, 53

[4] S. Young, "A review of large-vocabulary continuous-speech recognition," *Signal Processing Magazine, IEEE*, vol. 13, no. 5, pp. 45, 1996. 1, 52

[5] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, XA Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al., "The HTK book (for HTK version 3.4)," 2006. 2, 15, 63, 102

[6] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proceedings of ICASSP*. IEEE, 1992, vol. 1, pp. 13–16. 2

[7] N. Kumar, *Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition*, Ph.D. thesis, Johns Hopkins University, 1997. 2, 13, 14

[8] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *Proceedings of ICSLP*, 2004. 2, 48

[9] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. 4, 85

[10] Y.H. Sung and D. Jurafsky, "Hidden conditional random fields for phone recognition," in *Proceedings of ASRU*. IEEE, 2009, pp. 107–112. 4

[11] S. Liu and K.C. Sim, "Implicit trajectory modelling using temporally varying weight regression for automatic speech recognition," in *Proceedings of ICASSP*. IEEE, 2012, pp. 4761–4764. 6, 51, 53, 58, 72, 86, 96, 98

[12] S. Liu and K.C. Sim, "Temporally varying weight regression: A semi-parametric trajectory model for automatic speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 1, pp. 151 – 160, 2014. 6, 51, 86

[13] S. Liu and K.C. Sim, "Multi-stream temporally varying weight regression for cross-lingual speech recognition," in *Proceedings of ASRU*. IEEE, 2013, pp. 434–439. 6, 71, 89

[14] H. Hermansky, D.P.W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proceedings of ICASSP*. IEEE, 2000, vol. 3, pp. 1635–1638. 13, 14, 49, 80, 86

[15] M.J.F. Gales, "Semi-tied covariance matrices for hidden Markov models," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 272–281, 1999. 14, 54

[16] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proceedings of ICASSP*. IEEE, 2005, vol. 1, pp. 961–964. 14, 28, 51, 52, 53, 61, 62, 64

[17] D. Yu and M.L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proceedings of Interspeech*, 2011, pp. 237–240. 14, 86

[18] J-F Mari, Dominique Fohr, and J-C Junqua, "A second-order HMM for high performance word and phoneme-based continuous speech recognition," in *Proceedings of ICASSP*. IEEE, 1996, vol. 1, pp. 435–438. 16, 27

[19] L-M Lee and J-C Lee, "A study on high-order hidden Markov models and applications to speech recognition," in *Advances in Applied Artificial Intelligence*, pp. 682–690. Springer, 2006. 16, 27

[20] L-M Lee, "High-order hidden Markov model and application to continuous mandarin digit recognition," *Journal of Information Science and Engineering*, vol. 27, no. 6, pp. 1919–1930, 2011. 16, 27

[21] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970. 20

[22] J. Nocedal and S.J. Wright, *Numerical Optimization, 2nd Edition*, Springer, 2006. 23, 45, 62

[23] H. Gish and K. Ng, "A segmental speech model with applications to word spotting," in *Proceedings of ICASSP*, april 1993, vol. 2, pp. 447 –450 vol.2. 25

[24] H. Gish and K. Ng, "Parametric trajectory models for speech recognition," in *Proceedings of ICSLP*. IEEE, 1996, vol. 1, pp. 466–469. 25, 52

[25] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri, "A recognition method with parametric trajectory synthesized using direct relations between static and dynamic feature vector time series," in *Proceedings of ICASSP*. IEEE, 2002, vol. 1, pp. I–957. 25, 26

[26] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri, "Recognition method with parametric trajectory generated from mixture distribution HMMs," in *Proceedings of ICASSP*. IEEE, 2003, vol. 1, pp. I–124. 25, 26

[27] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri, "A theoretical analysis of speech recognition based on feature trajectory models," in *Proceedings of ICSLP*, 2004. 25, 26

[28] K. Tokuda, H. Zen, and T. Kitamura, "Trajectory modeling based on HMMs with the explicit relationship between static and dynamic features," in *Proceedings of Eurospeech*. Citeseer, 2003, pp. 865–868. 26, 52

[29] K. Tokuda, H. Zen, and T. Kitamura, "Reformulating the HMM as a trajectory model," in *Proceedings of Beyond HMM Workshop*, 2004. 26, 52

[30] H. Zen, K. Tokuda, and T. Kitamura, "An introduction of trajectory model into HMM-based speech synthesis," *Proceedings of ISCA SSW5*, pp. 191–196, 2004. 26

[31] H. Zen, K. Tokuda, and T. Kitamura, "A Viterbi algorithm for a trajectory model derived from HMM with explicit relationship between static and dynamic features," in *Proceedings of ICASSP*. IEEE, 2004, vol. 1, pp. I–837. 26

[32] C. Wellekens, "Explicit time correlation in hidden markov models for speech recognition," in *Proceedings of ICASSP*. IEEE, 1987, vol. 12, pp. 384–386. 27

[33] P.C. Woodland, "Hidden Markov models using vector linear prediction and discriminative output distributions," in *Proceedings of ICASSP*. IEEE, 1992, pp. 509–512. 27

[34] J.A. Bilmes, "Buried Markov models for speech recognition," in *Proceedings of ICASSP*. IEEE, 1999, pp. 713–716. 28

[35] A-V.I Rosti and M.J.F Gales, "Switching linear dynamical systems for speech recognition," Tech. Rep., 2003. 28, 52

[36] K.C. Sim and M.J.F. Gales, "Discriminative semi-parametric trajectory model for speech recognition," *Computer Speech & Language*, vol. 21, no. 4, pp. 669–687, 2007. 28, 51, 52, 53, 64

[37] K.C. Sim and M.J.F. Gales, "Discriminative semi-parametric trajectory model for speech recognition," *Computer Speech & Language*, vol. 21, no. 4, pp. 669–687, 2007. 28

[38] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977. 29

[39] V. Valtchev, J.J. Odell, P.C. Woodland, and S.J. Young, "MMIE training of large vocabulary recognition systems," *Speech Communication*, vol. 22, no. 4, pp. 303–314, 1997. 29

[40] V. Valtchev, J.J. Odell, P.C. Woodland, and S.J. Young, "Lattice-based discriminative training for large vocabulary speech recognition," in *Proceedings of ICASSP*. IEEE, 1996, pp. 605–608. 29

[41] B.H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *Signal Processing, IEEE Transactions on*, vol. 40, no. 12, pp. 3043–3054, 1992. 29

[42] P.S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "A generalization of the Baum algorithm to rational objective functions," in *Proceedings of ICASSP*. IEEE, 1989, pp. 631–634. 29

[43] Y. Normandin and S.D. Morgera, "An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition," in *Proceedings of ICASSP*. IEEE, 1991, pp. 537–540. 29

[44] D. Povey, "Discriminative training for large vocabulary speech recognition," *PhD Thesis, Cambridge University*, 2004. 30, 59, 60

[45] J-L. Gauvain and C-H. Lee, "Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994. 32, 37, 104, 105

[46] V. V Digalakis, D. Rtischev, and L.G. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 357–366, 1995. 32, 34, 37, 127

[47] P.C. Woodland, "Speaker adaptation for continuous density HMMs: A review," in *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001. 32

[48] M.J.F Gales and P.C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249–264, 1996. 32, 33, 37, 85, 104, 105, 112

[49] M.J.F. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 417–428, 2000. 32

[50] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 113–120, 1979. 36

[51] X. Cui, M. Afify, and Y. Gao, "MMSE-based stereo feature stochastic mapping for noise robust speech recognition," in *Proceedings of ICASSP*. IEEE, 2008, pp. 4077–4080. 36, 105

[52] J. Droppo, L. Deng, and A. Acero, "Evaluation of SPLICE on the Aurora 2 and 3 tasks," in *Proceedings of ICSLP*, 2002, pp. 29–32. 36, 105

[53] P. Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, 1996. 36, 105

[54] Veronique S., Hugo V.H., and Patrick W., "Accounting for the uncertainty of speech estimates in the context of model-based feature enhancement," in *Proceedings of ICSLP*, 2004. 36

[55] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with splice for noise robust speech recognition," in *Proceedings of ICASSP*. IEEE, 2002, vol. 1, pp. I–57. 37

[56] H. Liao and M.J.F. Gales, "Uncertainty decoding for noise robust speech recognition," in *Proceedings of Interspeech*, 2007. 37

[57] M.J.F. Gales and S.J. Young, "Cepstral parameter compensation for hmm recognition in noise," *Speech Communication*, vol. 12, no. 3, pp. 231–239, 1993. 37, 63, 95, 96

[58] A. Acero, L. Deng, T. Kristjansson, and J Zhang, "HMM Adaptation Using Vector Taylor Series for Noisy Speech Recognition," in *Proceedings of ICSLP*, 2000, pp. 869–872. 37, 38, 39, 63, 95, 96, 102, 106, 108, 121

[59] R.C. van Dalen and M.J.F. Gales, "Extended VTS for noise-robust speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 733–743, 2011. 37, 95, 96, 98

[60] M.J.F. Gales and S.J. Young, "A fast and flexible implementation of parallel model combination," in *Proceedings of ICASSP*, may 1995, vol. 1, pp. 133 –136. 37, 95, 96

[61] K.C. Sim and M-T. Luong, "A Trajectory-based Parallel Model Combination with a unified static and dynamic parameter compensation for noisy speech recognition," in *Proceedings of ASRU*. IEEE, 2011. 37, 95, 96, 102

[62] R.A. Gopinath, M.J.F. Gales, P.S. Gopalakrishnan, S. Balakrishnan-Aiyer, and J.A. Picheny, "Robust speech recognition in noise: Performance of the IBM continuous speech recognizer on the ARPA noise spoke task," in *Proceedings of ARPA Worksh. Spoken Lang. Syst. Tech.*, 1995, pp. 127–130. 40

[63] O. Kalinli, M.L. Seltzer, and A. Acero, "Noise adaptive training using a vector taylor series approach for noise robust automatic speech recognition," in *Proceedings of ICASSP*. IEEE, 2009, pp. 3825–3828. 40, 95, 99, 101, 106, 109

[64] A-r. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012. 40, 85

[65] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012. 40, 42, 43, 46, 85

[66] J. Pan, C. Liu, Z. Wang, Y. Hu, and H. Jiang, "Investigations of deep neural networks for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modelling," 2012. 40

[67] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," 1986. 41

[68] G.E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002. 43

[69] G.E. Hinton, L. Deng, D. Yu, G.E. Dahl, A-r. Mohamed, N. Jaitly, Andrew Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012. 44, 46, 85

[70] G.E. Hinton, S. Osindero, and Y-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006. 44

[71] I. Sutskever, J. Martens, G. Dahl, and G.E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of ICML*, 2013, pp. 1139–1147. 44

[72] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 1, pp. 213, 2002. 45

[73] J. Martens, "Deep learning via Hessian-free optimization," in *Proceedings of ICML*, 2010, pp. 735–742. 45

[74] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH*, 2011, pp. 437–440. 46

[75] B. Li and K.C. Sim, "Noise adaptive front-end normalization based on vector Taylor series for deep neural networks in robust speech recognition," in *Proceedings of ICASSP*, 2013, pp. 7408–7412. 46

[76] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proceedings of ASRU*. IEEE, 2011, pp. 24–29. 46, 86, 105

[77] Y. Miao and Metze F., "Improving Low-Resource CD-DNN-HMM using Dropout and Multilingual DNN Training," in *INTERSPEECH*, 2013. 46

[78] W. Byrne, P. Beyerlein, J.M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Peterek, J. Picone, D. Vergyri, and T. Wang, "Towards language independent acoustic modeling," in *Proceedings of ICASSP*. IEEE, 2000, vol. 2, pp. II1029–II1032. 47, 72

[79] R. Bayeh, S. Lin, G. Chollet, and C. Mokbel, "Towards multilingual speech recognition using data driven source/target acoustical units association," in *Proceedings of ICASSP*. IEEE, 2004, vol. 1, pp. I–521. 47

[80] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N.a Goel, M.n Karafiát, D. Povey, et al., "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *Proceedings of ICASSP*. IEEE, 2010, pp. 4334–4337. 47, 72

[81] H. Lin, L. Deng, D. Yu, Y-f. Gong, A. Acero, and C-H. Lee, "A study on multilingual acoustic modeling for large vocabulary asr," in *Proceedings of ICASSP*. IEEE, 2009, pp. 4333–4336. 47, 72

[82] L. Lu, A. Ghoshal, and S. Renals, "Regularized subspace gaussian mixture models for cross-lingual speech recognition," in *Proceedings of ASRU*. IEEE, 2011, pp. 365–370. 47, 72

[83] K.C. Sim and H. Li, "Context sensitive probabilistic phone mapping model for cross-lingual speech recognition," in *Proceedings of Interspeech*, 2008, pp. 2715–2718. 47, 72

[84] K.C. Sim and H. Li, "Robust phone set mapping using decision tree clustering for cross-lingual phone recognition," in *Proceedings of ICASSP*. IEEE, 2008, pp. 4309–4312. 47, 48, 72

[85] K.C. Sim, "Discriminative product-of-expert acoustic mapping for cross-lingual phone recognition," in *Proceedings of ASRU*. IEEE, 2009, pp. 546–551. 47, 48, 72

[86] V.B. Le and L. Besacier, "First steps in fast acoustic modeling for a new target language: application to vietnamese," in *Proceedings of ICASSP*, 2005, vol. 5, pp. 821–824. 47, 72

[87] O. Cetin, A. Kantor, S. King, C. Bartels, M. Magimai-Doss, J. Frankel, and K. Livescu, "An articulatory feature-based tandem approach and factored observation modeling," in *Proceedings of ICASSP*. IEEE, 2007, vol. 4, pp. IV–645. 48

[88] S. Thomas, S. Ganapathy, and H. Hermansky, "Cross-lingual and multistream posterior features for low resource lvcsr systems," in *Proceedings of Interspeech*, 2010. 48, 72

[89] A. Stolcke, F. Grézl, M-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proceedings of ICASSP*. IEEE, 2006, vol. 1, pp. I–I. 48, 72

[90] P. Lal, *Cross-lingual Automatic Speech Recognition using Tandem Features*, Ph.D. thesis, University of Edinburgh, 2011. 48, 49, 72

[91] M. Ostendorf and S. Roukos, "A stochastic segment model for phoneme-based continuous speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 1857–1869, 1989. 52

[92] M. Ostendorf, V. Digalakis, and O. A. Kimball, "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 360–378, 1995. 52

[93] A-V.I. Rosti and M.J.F. Gales, "Factor analysed hidden Markov models for Speech Recognition," *Computer Speech & Language*, 2004. 52

[94] L. Deng, X. Li, D. Yu, and A. Acero, "A Hidden Trajectory Model with Bi-directional Target-Filtering: Cascaded vs. Integrated Implementation for Phonetic Recognition," in *Proceedings of ICASSP*, 2005, pp. 337–340. 52

[95] J.A. Bilmes, "Buried Markov models for speech recognition," in *Proceedings of ICASSP*. IEEE, 1999, vol. 2, pp. 713–716. 52

[96] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent feature transforms for speech recognition," in *Proceedings of ICASSP*. IEEE, 2006, vol. 1, pp. 313–316. 52

[97] S.S. Kozat, K. Visweswariah, and R. Gopinath, "Feature adaptation based on gaussian posteriors," in *Proceedings of ICASSP*. IEEE, 2006, vol. 1, pp. I–I. 52, 62

[98] D. Povey and P.C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proceedings of ICASSP*. IEEE, 2002, vol. 1, pp. I–105. 52, 59, 61

[99] L. Si and R. Jin, "Adjusting mixture weights of gaussian mixture model via regularized probabilistic latent semantic analysis," in *Advances in Knowledge Discovery and Data Mining*, pp. 622–631. Springer, 2005. 53

[100] A. Wächter and L.T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. 53

[101] K.C. Sim and M.J.F. Gales, "Minimum phone error training of precision matrix models," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 882–889, 2006. 54

[102] M.J.F. Gales, "Model-based approaches to handling uncertainty," *Robust Speech Recognition of Uncertain or Missing Data-Theory and Applications. Springer, Berlin, Germany*, pp. 101–125, 2011. 63

[103] X. Cui, M. Afify, and Y. Gao, "Stereo-based stochastic mapping with discriminative training for noise robust speech recognition," in *Proceedings of ICASSP*. IEEE, 2009, pp. 3933–3936. 63

[104] N. Parihar and J. Picone, "Performance analysis of the Aurora large vocabulary baseline system," in *Proceedings of the European Signal Processing Conference*, 2004. 69, 101

[105] Y. Wang and Mark G.J.F. Gales, "Speaker and noise factorization for robust speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 7, pp. 2149–2158, 2012. 69, 106, 112, 119, 120, 122, 123, 124

[106] E. standard doc, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced frontend feature extraction algorithm; compression algorithms," Tech. Rep., 2003. 69

[107] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *Proceedings of Eurospeech*. Citeseer, 1999, vol. 99, pp. 495–498. 70

[108] D. Yu, L. Deng, P. Liu, J. Wu, Y. Gong, and A. Acero, "Cross-lingual speech recognition under runtime resource constraints," in *Proceedings of ICASSP*. IEEE, 2009, pp. 4193–4196. 72

[109] S.J. Young, J.J. Odell, and P.C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of HLT*, 1994, pp. 307–312. 76

[110] G. Wang and K.C. Sim, "An investigation of tied-mixture gmm based triphone state clustering," in *Proceedings of ICASSP*, 2012, pp. 4717–4720. 76

[111] "Phoneme recognizer based on long temporal context," in *Brno University of Technology, http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-author-context*. 79

[112] S. Liu and K.C. Sim, "On combining DNN and GMM with unsupervised speaker adaptation for robust automatic speech recognition," in *Proceedings of ICASSP*. IEEE, 2014, pp. –. 84, 90, 91, 106

[113] J-L Gauvain and C-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 291–298, 1994. 84, 85, 86

[114] O. Abdel-Hamid and H. Jiang, "Fast Speaker Adaptation of Hybrid NN/HMM Model for Speech Recognition based on Discrimative Learning of Speaker Code," in *Proceedings of ICASSP*. IEEE, 2013, pp. 7942 – 7946. 85, 105

[115] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on speaker code," in *Proceedings of ICASSP*. IEEE, 2014, pp. 6339–6343. 85

[116] Z-J. Yan, Q. Huo, and J. Xu, "A scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for LVCSR," in *Proceedings of Interspeech*. ISCA, 2013, pp. 1–1. 86

[117] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proceedings of ICASSP*. IEEE, 2007, vol. 4, pp. 757–760. 86

[118] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*. IEEE, 2013, pp. 55–59. 86

[119] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011. 86

[120] S. Liu and K.C. Sim, "An investigation of temporally varying weight regression for noise robust speech recognition," in *Proceedings of Interspeech*. ISCA, 2013, pp. 4761–4764. 94, 95, 106

[121] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 11, no. 6, pp. 568–580, 2003. 95

[122] S. Rennie, P. Dognin, and P. Fousek, "Robust speech recognition using dynamic noise adaptation," in *Proceedings of ICASSP*. IEEE, 2011, pp. 4592–4595. 95

[123] X. Cui and Y. Gong, "A study of variable-parameter Gaussian mixture hidden Markov modeling for noisy speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1366–1376, 2007. 95

[124] M. Fujimoto, S. Watanabe, and T. Nakatani, "Non-stationary noise estimation method based on bias-residual component decomposition for robust speech recognition," in *Proceedings of ICASSP*. IEEE, 2011, pp. 4819–4819. 95

[125] A. Ragni and M.J.F. Gales, "Derivative kernels for noise robust ASR," in *Proceedings of ASRU*. IEEE, 2011, pp. 119–124. 98

[126] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980. 102

[127] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proceedings of ICASSP*, 2013. 105, 106, 119, 120, 121, 122, 124, 127

[128] B. Li and K.C. Sim, "Noise adaptive front-end normalization based on vector taylor series for deep neural networks in robust speech recognition," in *Proceedings of ICASSP*. IEEE, 2013, pp. 7408–7412. 105

[129] B. Li and K.C. Sim, "Improving robustness of deep neural networks via spectral masking for automatic speech recognition," in *Proceedings of ASRU*. IEEE, 2013, pp. 279–284. 105, 127

[130] P. Vincent, H. Larochelle, Y. Bengio, and P-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of ICML*. ACM, 2008, pp. 1096–1103. 105

[131] D. Povey and K. Yao, "A basis representation of constrained mllr transforms for robust adaptation," *Computer Speech & Language*, vol. 26, no. 1, pp. 35–51, 2012. 119

[132] A. Narayanan and DL. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *OSU-CISRC-6/13-TR14*, 2013. 121, 122, 124, 127

[133] Bo Li and K Sim, "A spectral masking approach to noise-robust speech recognition using deep neural networks," pp. 1296–1305, 2014. 121, 122, 124

[134] Arun Narayanan and DeLiang Wang, "Joint noise adaptive training for robust automatic speech recognition," *Proceedings of ICASSP*, 2014. 121, 122, 124, 126

[135] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N.K. Goel, M. Karafiát, A. Rastrow, et al., "Subspace Gaussian mixture models for speech recognition," in *Proceedings of ICASSP*. IEEE, 2010, pp. 4330–4333. 127

[136] L. Wang and P.C. Woodland, "Discriminative adaptive training using the MPE criterion," in *Proceedings of ASRU*. IEEE, 2003, pp. 279–284. 127

[137] C.K. Raut, K. Yu, and M.J.F. Gales, "Adaptive training using discriminative mapping transforms," in *Proceedings of Interspeech*, 2008, pp. 1697–1700. 127

[138] F. Flego and M.J.F. Gales, "Discriminative adaptive training with vts and jud," in *Proceedings of ASRU*, 2009, pp. 170–175. 127

[139] LC. Yann and B. Yoshua, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, 1995. 127

[140] O. Abdel-Hamid, A-r Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of ICASSP*. IEEE, 2012, pp. 4277–4280. 127

[141] T.N. Sainath, A-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of ICASSP*. IEEE, 2013, pp. 8614–8618. 127

[142] A. Graves, A-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of ICASSP*. IEEE, 2013, pp. 6645–6649. 127

[143] G.E. Hinton, Srivastava N., A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580, 2012*. 127

[144] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proceedings of ICASSP*. IEEE, 2009, pp. 3761–3764. 127, 128

[145] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence discriminative training of deep neural networks," in *Proceedings of Interspeech*, 2013, pp. 2345–2349. 127, 128

Appendix A

Appendix

A.1 Jacobian Issue

Given $\mathbf{x} \in \mathbb{R}^n \sim \mathcal{N}(\mathbf{x}; \mu_\mathbf{x}, \mathbf{\Sigma_x})$, in order to be a valid probability formula, the following equation has to be satisfied

$$\int_{-\infty}^{+\infty} \mathcal{N}(\mathbf{x}; \mu_\mathbf{x}, \mathbf{\Sigma_x})\, \mathrm{d}\mathbf{x} = 1. \tag{A.1}$$

If a transformation is applied as $\mathbf{y} = f(\mathbf{x})$ where $f(\mathbf{x})$ is any differentiable function, Since the real variable is $\mathbf{x}$ rather than $\mathbf{y}$, the cumulative probability over all the possible $\mathbf{x}$ should be ONE. However, if the transformation is not just a simple global shifting, this constraint will no longer be guaranteed, that is

$$\int_{-\infty}^{+\infty} \mathcal{N}(\mathbf{y}; \mu_\mathbf{y}, \mathbf{\Sigma_y})\, \mathrm{d}\mathbf{x} \neq 1. \tag{A.2}$$

A similar equation to Eq-A.1 with respect to variable y can be written as

$$\int_{-\infty}^{+\infty} \mathcal{N}(\mathbf{y}; \mu_\mathbf{y}, \mathbf{\Sigma_y})\, \mathrm{d}\mathbf{y} = 1. \tag{A.3}$$

There is a strong dependency between these two variables, which can be written in derivative form:

$$\mathrm{d}\mathbf{y} = f'(\mathbf{x})\mathrm{d}\mathbf{x} \tag{A.4}$$

Since $\mathrm{d}\mathbf{y}$ is required to be positive for integral Eq-A.3, but not necessary for derivative (A.4), Eq-A.3 can be rewritten as:

$$\int_{-\infty}^{+\infty} \mathcal{N}(\mathbf{y}; \mu_\mathbf{y}, \mathbf{\Sigma_y}) |f'(\mathbf{x})|\, \mathrm{d}\mathbf{x} = 1. \tag{A.5}$$

where the absolute differential $|f'(\mathbf{x})|$ is also called Jacobian, denoted as $|\mathbf{J}_x|$, which can be a determinant if the function f is a multivariate to multivariate mapping. If the Jacobian $|\mathbf{J}_x|$ is a non-zero constant $\mathbf{J}$ independent of $\mathbf{x}$, one interesting conclusion about these two equations Eq-A.1 and Eq-A.5 can be drawn

$$\mathcal{N}(\mathbf{y}; \mu_{\mathbf{y}}, \mathbf{\Sigma}_{\mathbf{y}}) = \frac{1}{|\mathbf{J}|} * \mathcal{N}(\mathbf{x}; \mu_{\mathbf{x}}, \mathbf{\Sigma}_{\mathbf{x}}) \tag{A.6}$$

This observation is quite nice since the probability of the transformed variables can be calculated using the distribution of original space plus a Jacobian term $|\mathbf{J}|$ without explicitly calculating the distribution of the transformed variables. The reason why linear feature/model transformation is widely used for speech recognition is that the number of Jacobian is limited and can be pre-computed. However, the temporally varying model/feature transformation approaches have a temporally varying Jacobian, i.e. $\mathbf{J}_x$ depends on $\mathbf{x}$, if the transformation has dependency on the feature itself. Since the Jacobian needs a lot of computing resources, the requirement of too many evaluations of Jacobian can make the problem intractable. On one hand, the time-varying Jacobian term from the numerator and denominator in fMPE and pMPE objective function is canceled, so fMPE and pMPE does not have Jacobian issue. On the other hand, in order to estimate a time-varying feature transformation like fMPE using maximum likelihood estimation, explicitly evaluation of time-varying Jacobian terms will make it intractable for large scale problems.

A.2 Constraint Derivation for TVWR

This section provides the derivation of the constraints for the parameter c_{jm} and w_{jmi} in Eq-3.12 and Eq-3.13, respectively. Since c_{jm} is firstly introduced in Eq-3.1, c_{jm} should be constrained to make a valid probabilistic model for Eq-3.1 such that

$$\int_{\boldsymbol{\tau}_t} \int_{\mathbf{o}_t} p(\boldsymbol{\tau}_t, \mathbf{o}_t|j)\, \mathrm{d}\boldsymbol{\tau}_t \mathrm{d}\mathbf{o}_t = 1, \quad \forall j \tag{A.7}$$

$$\Longrightarrow \int_{\boldsymbol{\tau}_t} \int_{\mathbf{o}_t} \sum_{m=1}^{M} c_{jm} p(\boldsymbol{\tau}_t, \mathbf{o}_t|j, m)\, \mathrm{d}\boldsymbol{\tau}_t \mathrm{d}\mathbf{o}_t = 1, \quad \forall j \tag{A.8}$$

$$\Longrightarrow \sum_{m=1}^{M} c_{jm} \int_{\boldsymbol{\tau}_t} \int_{\mathbf{o}_t} p(\boldsymbol{\tau}_t, \mathbf{o}_t|j, m)\, \mathrm{d}\boldsymbol{\tau}_t \mathrm{d}\mathbf{o}_t = 1, \quad \forall j \tag{A.9}$$

Using the fact that the probability, $c_{jm} = P(m|j)$, is non-negative and

$$\int_{\boldsymbol{\tau}_t} \int_{\mathbf{o}_t} p(\boldsymbol{\tau}_t, \mathbf{o}_t|j, m)\, \mathrm{d}\boldsymbol{\tau}_t \mathrm{d}\mathbf{o}_t = 1 \tag{A.10}$$

leads to the constraints for c_{jm} such that

$$\sum_{m=1}^{M} c_{jm} = 1, \forall j \quad \text{and} \quad c_{jm} \geq 0, \forall j, m \tag{A.11}$$

Since $p(\boldsymbol{\tau}_t, \mathbf{o}_t|j, m)$ can be factorized into $p(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m)$ and $p(\mathbf{o}_t|j, m)$, the constraint in Eq-A.10 can be satisfied provided that the following are satisfied:

$$\int_{\mathbf{o}_t} p(\mathbf{o}_t|j, m)\, \mathrm{d}\mathbf{o}_t = 1, \quad \forall j, m \tag{A.12}$$

$$\int_{\boldsymbol{\tau}_t} p(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m)\, \mathrm{d}\boldsymbol{\tau}_t = 1, \quad \forall j, m, \mathbf{o}_t \tag{A.13}$$

Since $p(\mathbf{o}_t|j, m)$ is modelled by a GMM, the constraint in Eq-A.12 is satisfied. On the other hand, by applying the approximations in Eq-3.4,3.7,3.9, the constraint in Eq-A.13 is revised as:

$$\int_{\boldsymbol{\tau}_t} \tilde{p}(\boldsymbol{\tau}_t|\mathbf{o}_t, j, m)\, \mathrm{d}\boldsymbol{\tau}_t = 1, \quad \forall j, m, \mathbf{o}_t \tag{A.14}$$

$$\Longrightarrow \int_{\boldsymbol{\tau}_t} K_t \sum_{i=1}^{N} \tilde{p}(i|\boldsymbol{\tau}_t) w_{jmi}\, \mathrm{d}\boldsymbol{\tau}_t = 1, \quad \forall j, m \tag{A.15}$$

$$\Longrightarrow \sum_{i=1}^{N} w_{jmi} \int_{\boldsymbol{\tau}_t} p(\boldsymbol{\tau}_t|i)\, \mathrm{d}\boldsymbol{\tau}_t = 1, \quad \forall j, m \tag{A.16}$$

Since $\int_{\boldsymbol{\tau}_t} p(\boldsymbol{\tau}_t|i)\, \mathrm{d}\boldsymbol{\tau}_t = 1$ for all i and the probability, $w_{jmi} = P(i|j, m)$, is non-negative, the following constraints for w_{jmi} are obtained:

$$\sum_{i=1}^{N} w_{jmi} = 1, \forall j, m \quad \text{and} \quad w_{jmi} \geq 0, \forall j, m, i \tag{A.17}$$

Therefore, it is not necessary to constrain $\sum_m^M \tilde{c}_{jmt} = 1$ for all state j at each frame, t in order to ensure that $p(\boldsymbol{\tau}_t, \mathbf{o}_t|j)$ is a valid probability density function based on the simplifications. In other words, our instantaneous time-varying weights do not need to obey the sum-to-one constraint.

A.3 Solver for Discriminative Training of TVWR

In case of $C = 1$ in Eq-3.40, which is actually the most widely used setup, a fast implementation of searching the update solution can be found using Lagrange multiplier

method. The original constrained optimization problem is re-expressed to maximize following function:

$$y = \sum_{i=1}^{N} n_i \log x_i - \hat{d}_i x_i \tag{A.18}$$

where

$$\hat{d}_i = \frac{d_i}{\hat{x}_i}, \quad n_i \geq 0, \quad d_i \geq 0, \quad \hat{x}_i > 0, \forall i \tag{A.19}$$

subject to

$$\sum_{i=1}^{N} x_i = 1, \qquad x_i > 0 \, \forall i \tag{A.20}$$

Then, Lagrange function can be obtained as

$$L = \sum_{i=1}^{N} n_i \log x_i - \hat{d}_i x_i + \lambda(\sum_{i=1}^{N} x_i - 1) \tag{A.21}$$

In order to get the optimal solution, following equation system needs to be solved:

$$\frac{\partial L}{\partial x_i} = \frac{n_i}{x_i} - \hat{d}_i + \lambda = 0 \tag{A.22}$$

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{N} x_i = 1 \tag{A.23}$$

whose solution, λ^* is actually equivalent to the root of following function:

$$f(\lambda) = \sum_{i=1}^{N} \frac{n_i}{\hat{d}_i - \lambda} - 1 \tag{A.24}$$

Given the fact that $f'(\lambda) > 0$ and $f(\lambda) \in (-\infty, +\infty)$, there exists one and only one root of this function. Starting from $f(\lambda_0) < 0$, the root can be quickly found using Newton's method:

$$\lambda_0 = -\sum_{i=1}^{N} n_i \tag{A.25}$$

$$\lambda_{k+1} = \lambda_k - \frac{f(\lambda_k)}{f'(\lambda_k)} \tag{A.26}$$

Note that if $\hat{d}_i = 0$ for all i, then $\lambda^* = \lambda_0$. Therefore, optimal solution of this problem can be given as

$$x_i^* = \frac{n_i}{\hat{d}_i - \lambda^*} \tag{A.27}$$

A.4 Useful Matrix Derivatives

$$\frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a} \tag{A.28}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{b}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^T \tag{A.29}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X}^T \mathbf{b}}{\partial \mathbf{X}} = \mathbf{b}\mathbf{a}^T \tag{A.30}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{X}} = \frac{\partial \mathbf{a}^T \mathbf{X}^T \mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^T \tag{A.31}$$

$$\frac{\partial \mathbf{x}^T \mathbf{B} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{B} + \mathbf{B}^T)\mathbf{x} \tag{A.32}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X}^{-1} \mathbf{b}}{\partial \mathbf{X}} = -\mathbf{X}^{-T} \mathbf{a}\mathbf{b}^T \mathbf{X}^{-T} \tag{A.33}$$

$$\frac{\partial \log |\mathbf{X}|}{\partial \mathbf{X}} = \mathbf{X}^{-T} \tag{A.34}$$

$$\frac{\partial g(\mathbf{Y})}{\partial \mathbf{X}_{ij}} = \mathrm{Tr}\left[\left(\frac{\partial g(\mathbf{Y})}{\partial \mathbf{Y}} \right)^T \frac{\partial \mathbf{Y}}{\partial \mathbf{X}_{ij}} \right] \tag{A.35}$$