Robust Speech Recognition Against Unknown Short-Time Noise

by

CHAN, Yu Chung Arthur

B. Eng., Electrical and Electronic Engineering, The Hong Kong University of Science and Technology, Hong Kong, 1999

A Thesis Presented to
The Hong Kong University of Science and Technology
in Partial Fulfillment
of the Requirements for
the Degree of Master of Philosophy
in Electrical and Electronic Engineering

July 10, 2002, Hong Kong,

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

CHAN, Yu Chung Arthur

Robust Speech Recognition Against Short-Time Noise

by

CHAN, Yu Chung Arthur

This is to certify that I have examined the above MPhil thesis and have found that it is complete and satisfactory in all respects, and that any and all revisions required by the thesis examination committee have been made.

Dr. M. Siu, Thesis Supervisor

Dr. O. Au, Thesis Examination Chairman

Dr. B. Mak, Thesis Examination Examiner

Dr. W. H. Mow, Thesis Examination Examiner

Prof. H. S. Kwok, Acting Head of Dept. of EEE

Department of Electrical and Electronic Engineering

February 26, 2003

Acknowledgments

First, I would like to thank my thesis supervisor, Dr. Manhung Siu, for I believe that I have been his most troublesome student. I thank him for all his suggestions, his clear and clever explanation of difficult concepts, and for his patience for my blunders. In particular, I am grateful for the freedom that he has gave me during these years. Without him, this thesis would not exist.

I would also like to thank my mentor, Dr. Pascale Fung, in the spring semester of 1999/2000. She was also the supervisor of my Final Year Project. I would like to thank her because she gave me a memorable lesson regarding the meaning of 'professional'.

During the past years, I was happy to be one of the members of the project "Plaser", a software to learn pronunciation for high-school students. I would like to thank Dr. Brian Mak for his great management of the project.

The core program of this thesis, the frame-skipping Viterbi algorithm, makes use of routines in Plaser's recognizer. I would like to thank the main programmer, Wilson Tam, for his support in regard to the codes and for his kind suggestions.

My thanks also go to Wing Hui, for his help in solving many technical problems for me.

Further, I greatly benefitted from my interaction with members of the Language and Speech Technology Research (LASTRE) group at the Human Language Technology Center (HLTC) at Hong Kong University of Science and Technology (HKUST). There are too many people to mention individually, but I must thank Wing Hui (repeatedly) and Wilson Tam.

Many parts of my research discussed in this thesis have been submitted to various conferences. I would like to thank to Jacqueline Lo, Karen Leung and Jimmy Wong for their proofreading. (I am very sorry I was sometimes bossy!) I would also like to thank Chi-Yung Wang for his proofreading of the final version of this thesis.

This work was made possible by the excellent computing facilities in the LASTRE group. Thank you to the supporting team in the Computer Science

Department and the supporting team of LASTRE, Wilson Tam (again!), Ivan Chan and Chak-Fai Lee.

During the summer of 2001 I was lucky to be an intern at the Boston office of Speechworks Inc. I would like to thank Steven Lee who made my visit to the United States so enjoyable and rewarding.

My thanks also go to my family. During 2001 and 2002, my grandfather was suffering from cancer and he passed away on 26th March 2002. I should have been the one taking care of him. However, because of the understanding of other members of my family, I did not undertake this responsibility. My thanks go to my family for their support.

Last but not least, I would like to thank my girlfriend, Salina Ip. I would like to thank for her patience in listening to my 'great' ideas and her tolerance of my bad temper and lack of phone calls.

Robust Speech Recognition against Short-Time Noise

by

CHAN, Yu Chung Arthur

Electrical and Electronic Engineering Department

The Hong Kong University of Science and Technology

Abstract

The performance of automatic speech recognition (ASR) is known to degrade under noise corruption. Such means of corruption include additive noises such as air-conditioning noise, which corrupts across the whole speech waveform, short-time noises such as the slamming of a door or frame loss in network communication, which corrupts only a portion of the speech. Many previous researches focused on techniques which model the additive noise across the speech. However, for short-time noises that affect only a portion of speech, an alternative approach is to perform ASR on the unaffected (clean) parts of the speech signal.

In this thesis we propose a robust algorithm that automatically discards badly scored observations and which effectively performs ASR based on the clean observations only. Analogous to the use of the trimmed-mean as a robust estimate of the sample mean, we search for the state sequence with most likelihood that ignores K frames. Like using the Viterbi algorithm to find the maximum likely state sequence, we developed a dynamic algorithm, called the frame-skipping Viterbi algorithm (FSVA) to implement the proposed idea. To find the best number of skips, K, for different noise environments, we developed a method called log-likelihood ratio thresholding (LLRT) which gives a performance comparable to that obtained using the best value of K.

The FSVA is evaluated under two different noise environments, i) in GSM channels with white noise or comfort noise replacement and ii) in short-time additive noise in different conditions. In all cases, the FSVA outperforms the Viterbi algorithm.

The Viterbi algorithm has wide range of applications beyond ASR such as coding and pattern recognition tasks. These tasks can be potentially benefit from the FSVA.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic speech recognition (ASR) systems are computer systems that recognize spoken words. The development of such systems has now reached the point where applications such as dictation, command and control, and telephone voice inquiries have become possible. For example, in simple tasks such as connected digit recognition [52], an accuracy of 99% or more can be achieved. In larger scale tasks such as dictation, an ASR system can achieve an accuracy of up to about 95% [24]. If the task is under active research, a 10% relative improvement can be achieved per year because of the improvements in computational power and the advances in technology [23].

Ideally, ASR systems should show an acceptable performance in any situations. However, as it has been found after years of research, there are many factors that significantly affect the accuracy of an ASR system. Some of these factors are,

- **Speaker Variations**

  Speakers of different genders and ages utter speech with different characteristics.

- **Pronunciation Variations**

  A word may be pronounced differently. The pronunciations depends on the speaker's nationality and age.

- **Environment Variations**

  Any variation in the environment will significantly change the characteristics of speech.

In general, these variations should be able to be handled in an ASR system. The ASR system should be insensitive or *robust* to them. In our research, we endeavored to incorporate environment robustness to the ASR system.

There are many environmental variation, for example, the presence of background noise such as air-conditioning or background speech. Noises can also be unexpected in nature, for instance, the slam of a door and a telephone ringing. In general, conditions in reality and those in the laboratory always have some differences and we call these **environmental mismatches**. Designing an ASR system that can deal with environmentally mismatches is the subject of research in **noisy speech recognition** or **robust speech recognition**. Designing such a computer system is, arguably, a major challenge in the field of speech recognition.

Environmental mismatches can be broadly categorized into two types. One is additive in nature and may be caused by the background noises. This is called the **additive noise**. The other includes those differences between the recording channels. For example, the difference between microphones in laboratory and operating environments. This is called **channel noise**. In our research we focused on additive noise. In particular, we endeavored to deal with noises that are short-time in nature. Examples of such noises include door-slams, the click sounds of a keyboard and frame loss in voice communication in network environments. These noises have a special characteristic in that they only affect a portion of the speech signal. Hence, they are also called **partial and temporal corruption** [42].

Many previous researches on additive noise focused on techniques that model the noisy speech statistics. For example, Gales suggested the use of the parallel model combination (PMC) that first gathered the noisy speech statistics and used them to create the statistics of the noisy-speech signal [19]. However, an alternative approach for dealing with short-time noise is to extract the information from the unaffected part of speech [42]. This approach is actually more attractive than a noise modeling approach such as PMC because even if enough noise samples are gathered, the noisy speech model does not match the unaffected part of the speech signal.

Extracting information from the unaffected part of the speech signal raises the following interesting question,

- **Could the contributions of those corrupted frames in the decision making process of the speech recognition be omitted?**

This idea is closely related to a theory called **missing feature theory** [36] in which the focus is on how corrupted data should be tackled. Based on this theory, we developed a technique called the frame-skipping Viterbi algorithm (FSVA) to handle short-time corruption. In brief, FSVA is a dynamic programming algorithm which omits the contributions of k worst performing frames in the decision process of speech recognition. In this thesis, we present the theory and discuss the evaluation result of the algorithm.

1.1 Organization

Our proposed method is a modification of the conventional Hidden Markov Model (HMM)-based approach of speech recognition. Hence, in Chapter 2, we first describe the general theory of the HMM, and the application of the HMM in speech recognition.

Some general techniques in noisy speech recognition are presented in Chapter 3. In particular, we discuss techniques that dealt with additive noise in general. We then narrow down our discussion to those techniques that apply the missing feature theory. Finally, we discuss other techniques that could deal with short-time noise that partially and temporally corrupted the speech signal.

In Chapter 4 the theory of our proposed method, the frame-skipping Viterbi algorithm (FSVA) is introduced. We discuss the motivation of our algorithm and its relation to techniques introduced in Chapter 3. We also discuss two related implementation schemes. Both allow the skipping of a known number of frames in a standard Viterbi algorithm.

We evaluate the FSVA in regard to short-time noise in the rest of the thesis. Our research in respect to the FSVA was divided to two stages, In Chapter 5 we deal with a simple task in which speech utterances' frames are randomly replaced by Gaussian noise. This task simulates frame loss which occurs during speech transmission in network environment such as Global System Mobile (GSM) and Voice Over IP (VoIP).

The experiments reported in Chapter 5 actually gave us an understanding of the strengths and limitations of our proposal. We realized that the number of skips in FSVA was a crucial factor with respect to performance. This motivated us to develop a mechanism that automatically determines the number of skips called log-likelihood ratio thresholding (LLRT). Another issue that the FSVA caused the increase in the memory use compare to the standard Viterbi algorithm. We developed an algorithm, the simplified frame-skipping Viterbi algorithm (SFSVA) to tackle this problem. This is described in Chapter 6.

Using the LLRT and the SFSVA discussed in Chapter 6, we evaluate the FSVA in two more complex tasks in Chapter 7. The goal of the first task was to gain a further understand the properties of the FSVA. We evaluated FSVA on short-time additive noise with different amplitudes, durations, lengths. The goal of the second task was to verify whether the FSVA is useful in more realistic situations of speech transmission. In real-life speech transmission, for example in a GSM environment, the coder-decoder (codec) replaces the detected corrupted frame with predefined comfort noise. It became a question of whether the FSVA can ensure a better performance in this situation. We evaluate the FSVA on utterances transmitted through the GSM environment with frame loss replaced by comfort noises. This is described in Chapter 7.

In Chapter 8, we summarize the results, and discuss our contributions and possible directions for the future.

Chapter 2

Review of the HMM-based Speech Recognition System

We mentioned in Chapter 1 that our proposed technique modifies the conventional HMM-based speech recognizer and allows it to cope with short-time noise. To better explain our technique we will first discuss the general theory of speech recognition and HMM-based speech recognition.

In Section 2.1 of this chapter we discuss a statistical formulation of speech recognition. Two important ingredients in speech recognition, acoustic modeling (Section 2.1.1) and language modeling (Section 2.1.2) are introduced. In Section 2.2 we describe the evaluation metric of ASR systems.

Speech recognition begins with the mapping process of the speech samples to the observation sequence through the feature extraction process. One such mapping, mel frequency cepstral coefficient (MFCC) and is reviewed in Section 2.3. Though feature extraction is not the central theme of this work, the discussion is useful for understanding of some noisy techniques described in Chapter3.

In the rest of this chapter we describe the use of the HMM and its application in speech recognition. The basic definition and its related problems are discussed in Section 2.4. Some simple applications of the HMM in speech recognition such as isolated word recognition and connected word recognition are described in Section 2.5.

2.1 Statistical Formulation of Speech Recognition

The statistical formulation [26] of speech recognition is described in this section.

Let O be the evidence that represents the speech signal. Without loss of generality, we assume can that O is a sequence of symbols, $o_1, \ldots, o_n$, where o_n represents the evidence of the speech signal at time n. The process of transforming the speech signals into the evidence sequence is known as feature extraction or speech parametization. We describe this later in Section 2.3

Let W be a word sequence $w_1, \ldots, w_m$ where $w_i \in \mathcal{W}$ and $\mathcal{W}$ is a fixed and known set of words.

One common way to describe the task of speech recognizer is to find the most likely word sequence, W, given an observation sequence O.[1]

Mathematically, this means that the recognizer searches a word sequence $\tilde{W}$ such that,

$$\tilde{W} = \arg\max_{W} P(W|O). \tag{2.1}$$

The above equation can be rewritten using Bayes' formula,

$$\tilde{W} = \arg\max_{W} \frac{P(W)P(O|W)}{P(O)}. \tag{2.2}$$

where $P(W)$ describes the probability of the word sequence, W, is uttered. $P(O|W)$ describes the probability of the observation sequence, O, is observed if W is uttered.

Since $P(O)$ is independent of W, it can be omitted in the max operation. The recognizer thus searches for the best word sequence based on the criterion,

$$\tilde{W} = \arg\max_{W} P(W)P(O|W). \tag{2.3}$$

The above formulation implies that there are two major components in an ASR system. They are the **acoustic model**, $P(O|W)$, and the **language**

[1] This is not the only possible formulation for speech recognition. We chose this approach mainly because it is a common formulation. There are other formulations. For example, one may describe the task of a speech recognizer as minimizing the Bayes' Risk, instead of maximizing the likelihood[28].

model, $P(W)$. They are very important ingredients of a speech recognizer and we discuss them in the rest of this section.

2.1.1 Acoustic Model

Variability is an intrinsic characteristic of speech signals. The speech signals of different utterance of the same word sequence differs because of many factors such as the age, gender, speaking style and the accent of the speakers. Therefore, capturing the variability of speech is a fundamental problem when building a good acoustic model.

Usually, the result of learning is encoded as a mathematical model. The **parameters estimation** problem or training problem focuses on how we can learn the characteristics of speech given some training utterances.

After parameters estimation, a model that represents the statistical properties of speech sound is obtained. Given an input utterance, the **pattern matching** problem or testing problem searches for a word sequence, $\tilde{W}$, that best matches on the input utterance.

The model in question can have many possible representations. For example, the artificial neural network (ANN)[35] and the hidden Markov model (HMM) are two such representations. As we will propose techniques which are modified from the HMM-based method, we will describe the HMM and its application in acoustic modeling in more detail in Section 2.4 and Section 2.5.

2.1.2 Language Model

The language model, or the prior probability, $P(W)$, is another important part of statistical speech recognition. This can be understood by the fact that not every word sequence is equally likely to be uttered by a speaker. This prior probability can be represented as, [2]

$$P(W) = P(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_1, \ldots, w_{i-1}). \tag{2.4}$$

In our research, our proposed technique was independent of whether or not the language model is used. To simplify our discussion, we make the unigram assumption that the occurrence of a word does not depend on previous words. Mathematically,

$$P(W) = P(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i). \tag{2.5}$$

2.2 Evaluation of Speech Recognition Systems

The simplest way of defining the accuracy of a recognizer would be,

$$Acc = \frac{C}{N} \times 100\%. \tag{2.6}$$

where C is the number of correct words and N is the total number of words. This use of this simple definition may cause a problem in connected word recognition because the recognizer word sequence may have a length different to that of the reference word sequence. For example,

$$\begin{aligned} S_1 &: \text{abcd} \\ S_2 &: \text{bedfh} \end{aligned}$$

where each character represents a single word. Hence, a word sequence-matching algorithm is first applied to create a word-to-word alignment of the two string sequences. One such algorithm can be found in [24]. In our research, we used the word sequence matching algorithm and the default parameters defined in HTK [67]. The matching algorithm aligns the two word sequences,

[2] We applied Bayes' rule in Equation 2.4. Actually, Bayes' rule provides many possible decompositions of $P(W)$. However, it is more intuitive to assume that the occurrence of a word depends only on words that occurred previously.

$$
\begin{array}{rcl}
a & \longmapsto & \_ \\
b & \longmapsto & b \\
c & \longmapsto & e \\
d & \longmapsto & d \\
\_ & \longmapsto & f \\
\_ & \longmapsto & g
\end{array}
$$

where $\_$ is the null string. This example shows that there are three types of error.

1. Substitution errors

   For example, c is substituted by e.

2. Insertion errors

   For example, f and g.

3. Deletion errors

   For example, a.

In order to take account of the insertion and deletion errors into computation of accuracy, the word accuracy is defined as,

$$A = \frac{H - S - I - D}{H} \times 100\%, \tag{2.7}$$

where

H is the total number of words in the reference,

S is the number of substitution errors,

I is the number of insertions errors and

D is the number of deletions errors.

This definition is used throughout this thesis.

2.3 Feature Extraction and the Mel Frequency Cepstral Coefficient (MFCC)

It was assumed in Equation 2.3 that some kind of acoustic evidence, O, which can represent the speech signal exists, the process of transforming the speech signal into this acoustic evidence is known as feature extraction. There are many techniques which can be used for feature extraction. Examples include the Linear Predictive Coding (LPC) [52] and the Mel-Frequency Cepstral Coefficient (MFCC) [15]. The MFCC is commonly used in many ASR systems.

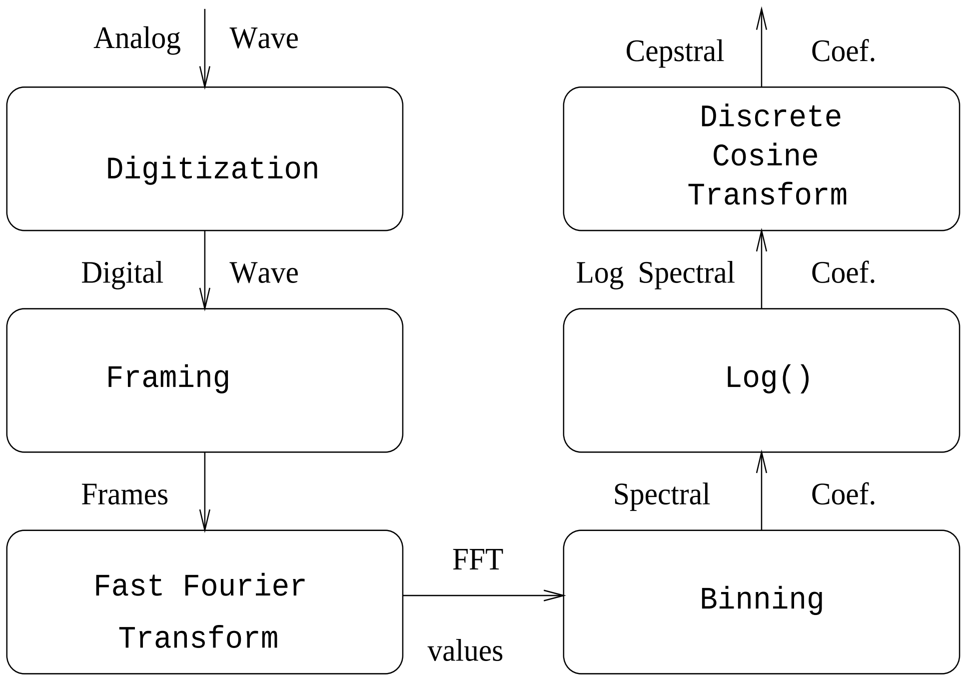

Figure 2.1: The block diagram for the MFCC

The typical process of generating the MFCC is shown in the block diagram in Figure 2.1. The analog speech signal is first digitized and forms $O(t)$. Each sample of the digitized speech signal is first pre-emphasized using a pre-emphasis window in the form of,

$$O(t)' = O(t) - 0.97O(t-1),$$

A moving Hamming window is then applied to the digitized speech signal.

Its weight can be expressed as,

$$w(t) = (0.54 - 0.46\cos(\frac{2\pi t}{T-1})).$$

Here, T is the length of one frame in the speech signal. Frames are then resulted in this process.

After this pre-processing stage, each frame is then processed in the following manner. First of all, the Fast Fourier Transform (FFT)[49] is then applied to each frame. The FFT coefficients $O(f)$ are then combined in order to reduce the size of the feature vector. This is done by applying a set of frequency banks (**bins**) to the coefficients such as the mel-scaled bins. This set of bins motivated by human perception research [15]. The process is also called binning and in this process the spectral coefficients are smoothed. This can be interpreted as a kind of smoothing to the spectral coefficient.

$$Mel(f) = 2595\log_{10}(1 + \frac{f}{700}).$$

The final linear spectral coefficients have the form of

$$O_b(\tau) = \sum_{f=1}^{F} w_b(f)^2 O(f),$$

where $w_b(f)$ is the weight associated with bin b, F is the highest bin component. Here $O_b(\tau)$ is the function of the binned frequency, τ.

The log spectral coefficients are then obtained by taking the logarithm of the spectral coefficients,

$$O_b^{log}(\tau) = \log(O_b(f, \tau)).$$

The cepstral parameters are then obtained by taking the discrete cosine transform (DCT) to the log spectral parameters,

$$O_i(v) = \sum_{b=1}^{B} O_b^{\log}(\tau)\cos(i(b-0.5)\pi/B),$$

where B is the number of bins after binning. Here v is "quefrency"[49] and $O_i(v)$ is the function of "quefrency".

Finally, a cepstral domain filter (a lifter) is then applied to the cepstral coefficients. The final output is the MFCC of one frame.

2.4 Theory of the Hidden Markov Model

The general theory of the HMM can be found in [26], [52]. In this section we focus on how the HMM is used in pattern recognition in general. In Section 2.4.1 we give a definition of the HMM. We describe the main problem encountered when using the HMM in speech recognition and how the probability of an observation sequence given the HMM is computed. This problem can be exactly solved by using the forward-backward algorithm [52] (Section 2.4.2) or approximately solved by using the Viterbi algorithm (Section 2.4.3). The Viterbi algorithm is commonly used because of its ability to trace back the maximum likely state sequence. We, thus, describe it in more detail [3].

Perhaps the most important issue in applying the HMM is to learn the HMM's parameters. This is, however, not the concern of this thesis. Hence, we summarize this with a few estimation formulas in Section 2.4.4.

2.4.1 The Hidden Markov Model (HMM)

A HMM consists of following components[4],

- **State Space**

  The set of all states in the HMM. The states are denoted as $q_1, q_2, \ldots, q_N$ where N is the number of states in the HMM.

- π : **Initial probability**

  A vector of N initial probabilities of starting in the state i for $1 \leq i \leq N$.

- A : **Transition probability**

  It can be represented as a $N \times N$ matrix. The (i, j)-th element of the matrix, a_{ij}, specifying the probability of transiting from the i-th state to the j-th state for $1 \leq i \leq N, 1 \leq j \leq N$.

[3] The statement is not true for all paradigms of searching. For example, in tree search [26], the computation of total likelihood can be easily incorporated in the search.

[4] There are alternative definitions of the HMM that attach observation probabilities to the transition arcs rather than the states.[26] is one of them.

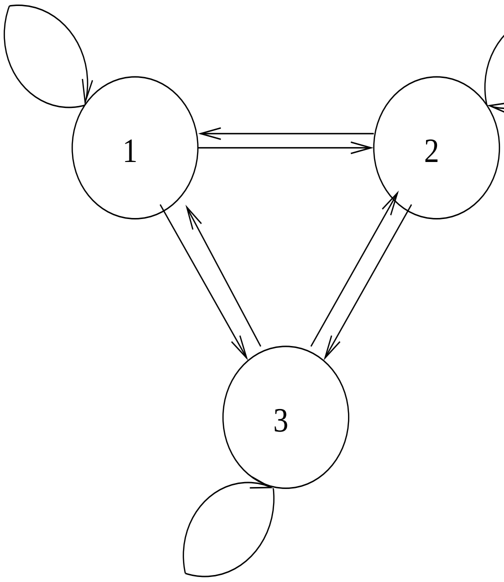

Figure 2.2: An ergodic HMM.

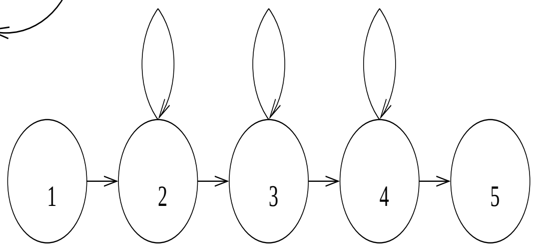

Figure 2.3: A left-to-right HMM.

- B : **Observation probability**

  A vector of N distributions, in which $b_j(o_t) = p(x|q_t = j)$ is usually a multidimensional distribution that describes the probability of observation vector x given it is generated by state j for $1 \leq j \leq N$.

We can express a HMM as a directed graph. For example, in Figure 2.2, a fully connected HMM is shown. It is also called the **ergodic HMM**. In Figure 2.3 a **left-to-right HMM** is shown.

Apart from speech recognition, the HMM has also been applied to several other applications. Examples of these applications include hand-written character recognition [9] and face recognition [48].

To apply the HMM in pattern recognition tasks, the first task is to calculate $P(O|\lambda)$ where O is the observation sequence and λ is the HMM. This requires one to calculate

$$\sum_{Q \in \tilde{Q}} P(O, Q|\lambda), \tag{2.8}$$

where $\tilde{Q}$ is the set of all possible state sequences. The above expression represents the sum of probability for all possible state sequence. This problem can be efficiently solved by the forward-backward algorithm [52].

2.4.2 Forward Backward Algorithm

If we try to solve this problem by exhaustively searching all possible state sequences, the computation is intractable.

$$\begin{aligned} P(O|\lambda) &= \sum_Q P(O,Q|\lambda), \\ &= \sum_Q P(O|Q,\lambda)P(Q|\lambda), \\ &= \sum_Q P(Q|\lambda)(\sum_{t=1}^{T} P(o_t|q_t,\lambda)). \end{aligned} \tag{2.9}$$

The second term in the product is just $b_{q_t}(o_t)$, while the first term

$$\begin{aligned} P(Q|\lambda) &= P(q_1|\lambda)\sum_{t=1}^{T-1} P(q_{t+1}|q_t,\lambda), \\ &= \pi_1 \prod_{t=1}^{T-1} a_{t,t+1}. \end{aligned} \tag{2.10}$$

Rearrange the terms and we will get

$$P(O|\lambda) = \sum_Q \pi_1 \prod_{t=1}^{T-1} a_{t,t+1} b_{q_t}(o_t). \tag{2.11}$$

However, enumerating all possible state sequences requires computation in the order of $O(n^T)$. This is not plausible for any useful value of T. Fortunately, many computations are actually redundant. This implies there can be a significant saving of computations and is basically the idea behind the forward-backward algorithm.

We first establish the recursive relation. Defining the **forward probability** in state i at time t as,

$$\alpha_i(t) = P(o_1 \dots o_{t-1}, q_t = i|\lambda), \tag{2.12}$$

and we denote the parameter set of HMM $[A, B, \pi]$ as λ for simplicity.

From this definition, we can construct $\alpha_i(t+1)$ very easily

$$\alpha_i(t+1) = (\sum_{j=1}^{N} a_{j,i}\alpha_j(t))b_i(o_{t+1}). \tag{2.13}$$

Using this recursive relation, we can derive a fast algorithm to compute $P(O|\lambda)$ (in our case $W = \lambda$).

1. **Initialization** $\alpha_i(1) = \pi_i$ for $1 \leq i \leq N$.
2. **Forward Calculation**

   for t from 1 to $T-1$

   calculate $\alpha_i(t+1)$ base on formula

   $$\alpha_i(t+1) = (\sum_{j=1}^{N} a_{j,i}\alpha_j(t))b_j(O_{t+1}). \tag{2.14}$$

3. **Termination** $P(O|W)$ can be calculated as

   $$P(O|W) = \sum_{j=1}^{N} \alpha_j(T) \tag{2.15}$$

This algorithm is known as the **forward algorithm**.

The above calculation can also be done in the descending order of time. If we defined the **backward probability** in state i at time t as,

$$\beta_i(t) = P(o_{t+1} \dots o_T | q_t = i, \lambda). \tag{2.16}$$

The proof can be found in [52].

As the forward-backward algorithm requires computation of the likelihood for all possible state sequences, some very unlikely state sequences are also taken into account in the computation. In practice, one can just calculate the path with the maximum likelihood to approximate $P(O|\lambda)$. Mathematically,

$$P(O|\lambda) \approx \max_Q P(O, Q|\lambda). \tag{2.17}$$

In the next section, we describe an efficient algorithm known as the Viterbi Algorithm for calculating $\max_Q P(O, Q|\lambda)$.

2.4.3 Viterbi Algorithm

Given an observation sequence O, and model $\lambda = [A, B, \pi]$ with N states, computation of $P(O|\lambda)$ can be approximated as

$$P(O|\lambda) \approx \max_{Q} P(O, Q|\lambda), \tag{2.18}$$

and the most-likely state sequence $\tilde{Q}$ can be found by,

$$\tilde{Q} \quad = \quad \arg\max_{Q} P(O, Q|\lambda).$$

Enumerating all possible state sequences and computing the likelihood for all of them is computationally intensive. To solve this efficiently, we can carry out the computation can be carried out using the following procedure.

We first define

$$\delta_i(t) = \max_{Q} P(o_1 \dots o_{t-1}, q_1 \dots q_{t-1}|q_t, \lambda). \tag{2.19}$$

We can easily construct $\delta_i(t+1)$ from $\delta_i(t)$,

$$\delta_i(t+1) = (\max_{j} a_{j,i}\delta_i(t))b_j(o_{t+1}). \tag{2.20}$$

The Viterbi algorithm is given as follows:

1. **Initialization** $\delta_i(1) = \pi_i$ for $1 \leq i \leq N$.

2. **Recursion**

   for t from 1 to $T-1$

   calculate $\alpha_i(t+1)$ base on formula

$$\delta_i(t+1) = (\max_{j=1}^{N} a_{j,i}\delta_j(t))b_i(o_{t+1}), \tag{2.21}$$

$$\psi_i(t+1) = \arg(\max_{j=1}^{N} a_{j,i}\delta_j(t))b_i(o_{t+1}). \tag{2.22}$$

3. **Termination** $P(O|\lambda)$ can be calculated as

$$P(O|\lambda) = \max_{j=1}^{N} \delta_j(T). \tag{2.23}$$

4. **Back-Tracking**

$$\begin{aligned} \tilde{q}_t &= \psi_{t+1}(\tilde{q}_{t+1}) \quad \text{for } 1 \leq t \leq T, \\ \tilde{Q} &= (\tilde{q}_1 \ldots \tilde{q}_T). \end{aligned}$$

Notice that in order to keep track of the maximum likely path, we introduce a backtracking pointer ψ in each state at each time. It stores the state index of the most likely state of the previous time that updates a state.

2.4.4 Learning HMM Parameters and the Baum-Welch Algorithm

In previous sections we assumed that the parameters of the HMM were learned beforehand. There are several ways to learn these parameters from data. One common way is the so called the Maximum Likelihood (ML) approach, which applies the Baum-Welch algorithm in an iterative manner. Other approaches such as minimum classification error (MCE) and maximum mutual information (MMI) [18] are also commonly used in speech recognition. In this section we give the expression for the transition probabilities and the parameters of observation probabilities assuming that it is modeled by Gaussian distributions. The proofs can be found in standard textbook such as [24].

$$\begin{aligned}
\tilde{a}_{ij} &= \frac{\sum_{t=1}^{T} \alpha_{t-1}(i) a_{ij} b_j(o_t) \beta_t(j))}{\sum_{t=1}^{T} \sum_{l=1}^{N} \alpha_{t-1}(i) a_{il} b_l(o_t) \beta_t(l)}, \\
\mu_j &= \frac{\sum_{t=1}^{T} \alpha_t(i) \beta_t(i) o_t}{\sum_{t=1}^{T} \alpha_t(i) \beta_t(i)}, \\
\Sigma_j &= \frac{\sum_{t=1}^{T} \alpha_t(i) \beta_t(i)(o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} \alpha_t(i) \beta_t(i)}.
\end{aligned}$$

2.5 Application of the Hidden Markov Model in Speech Recognition

We described the theory of the HMM in Section 2.4. We will now explain how it is applied in speech recognition. The first issue we are interested in is the choice of observation probability distribution. In Section 2.5.1 we discuss the possible observation probability distribution and the definition of the mixture distributions, which is widely used in speech recognition. In the previous section, we described how to evaluate $P(O|\lambda)$. If we assume that a word is modeled by a HMM, we can readily perform the tasks of **isolated word recognition**. We briefly describe this task in Section 2.5.2 and how we can extend the HMM framework to perform tasks such as **connected word recognition**, which assumes a word string, instead of a single word is uttered by the speaker[5] in Section 2.5.3.

[5] The topic of large vocabulary continuous speech recognition (LVCSR) because it is out of the scope of our discussion. However, the result in this thesis also applies to LVCSR. Please refer to [26] about this topic.

2.5.1 Observation Probability Distribution Used in Speech Recognition

We will now describe several observation distributions, $b_j(o_t)$, commonly used in speech recognition [24].

- **Gaussian Distribution**

  This is also known as normal distribution and its multidimensional form is

  $$g(x) = \frac{1}{(2\pi)^{\frac{d}{2}}(|\Sigma|)^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)} \tag{2.24}$$

  The definition of Gaussian distribution depends on the following parameters,

  - μ **, the mean vector**
  - Σ **, the covariance matrix**

- **Gaussian Mixture Model (GMM)**

  Usually, real-life data are not normally distributed. Therefore, instead of using single Gaussian distribution for modeling, a weighted sum of Gaussian distributions is used to model the observation distribution. Mathematically, the observation distribution, $G(x)$, can be expressed as,

  $$G(x) = \sum_{x=i}^{N} w_i g_i(x) \tag{2.25}$$

  Here w_i is the weight of the i-th Gaussian distribution and $\sum w_i = 1$.

  Other distributions are also used in speech recognition. For example, the use of the homogeneous alpha mixture to model observation distribution is suggested in [12].

The following discussion is based on the GMM that is commonly used in the ASR community.

2.5.2 Isolated Word Recognition

In isolated word recognition, the speaker utters a single word at each utterance. If we assume that every word in the vocabulary can be modeled by a HMM, λ_w, the decision process of finding the best word is,

$$\tilde{w} = \arg\max_{w} P(O|\lambda_w). \tag{2.26}$$

This can be approximated as

$$\tilde{w} = \arg\max_{w}(\max_{Q} P(O, Q|\lambda_w)). \tag{2.27}$$

Hence, we can use the Viterbi algorithm mentioned in Section 2.4.3 to calculate $\max_Q P(O, Q|\lambda_w)$.

2.5.3 Connected Word Recognition

In connected word recognition (CWR), the speaker utters a sequence of word with unconstrained length, instead of a single word. The formulation of HMM can be extended to perform this task. The idea is to construct a HMM that represents all possible sequences of words. If the effect of the language model is omitted as we discussed in Section 2.1.2, we can construct this model for all possible word sequences as in Figure 2.4. Notice that every word in the diagram is represented by a HMM. We call the constructed HMM as λ_{cont}.

Hence, the recognition process can be regarded as finding the best state sequence that best describes the acoustic evidence given λ_{cont}. Mathematically, the best word sequence, $\tilde{W}$, is given by, [6]

$$\begin{aligned} \tilde{W} &= \arg\max_{W}(\sum_{Q} P(O, Q|\lambda_{cont})), && (2.28) \\ &\approx \arg\max_{Q} P(O, Q|\lambda_{cont}). && (2.29) \end{aligned}$$

[6] In the second equality, we made an assumption that the best word sequence can be represented by the best state sequence.

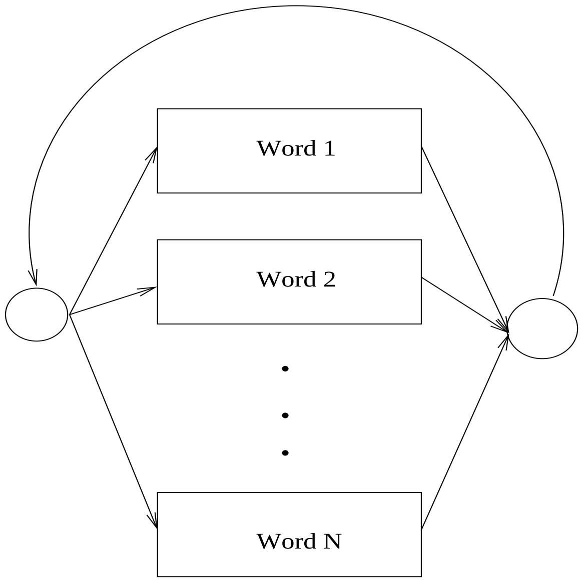

Figure 2.4: A word loop HMM.

Chapter 3

Speech Recognition in Mismatched Conditions

In the last chapter, we discussed the basic theory of HMM-based ASR systems. These systems achieve satisfactory performance provided that the operating (or testing) condition matches the laboratory (or training) condition. We call this **matched condition**. In many applications such as dictation, the ASR system operates in an office environment. It can be expected that background noises such as air-conditioning noise and fan noise existed in the environment. In such cases, these background noises can be injected into the training utterances to create a condition that matches the operating condition in training. This method also works for fixed-line telephone speech recognition in which headsets and microphones have become standardized in these days. Creating matched conditions becomes possible in this case.

In many cases, matching the training and testing conditions is very difficult, if not impossible. One typical condition nowadays is the wireless communication environment. The end-users in such an environment can be located in positions with various of background noises. Moreover, different kinds of hand-sets use different types of microphones. These microphones can show various channel responses. Hence it is difficult to incorporate all these conditions in training. In our research we focused on this situation, and called it **mismatched conditions**.

Robust speech recognition is the study of techniques that can deal with mismatched conditions in speech recognition.

There are many possible causes of mismatched conditions and we summarize them as follows. First of all, the training environment and the operating environment may be affected by background noise. For example, air-conditioning noise exists in the training environment and fan noise exists in the operating environment. Another possible variation is differences in microphones and transmission channels. In environment in which noise is correlated with speech such as reflection and reverberations, the noise changes as the speech characteristics and the environment change. In addition to these effects directly coming from the acoustic environment, speakers alter their speech pattern in response to stress or background noise. This is known as the Lombard effect [29]. A general framework to describe all the noise sources of mismatches can be found in [19] or [21]. The observed corrupted signal in the mismatched environment, $s'(t)$ can be described as,

$$s'(t) = (s(t)^{n_1(t)}_{Lombard} + n_1(t)) * h_{mike}(t) + n_2(t)) * h_{chan}(t) + n_3(t),$$

where $n_1(t)$ is the background noise, h_{mike} is the impulse response of the microphone, $n_2(t)$ and $h_{chan}(t)$ are respectively, the additive noise and impulse noise of the transmission channel. $n_3(t)$ is the additive noise introduced by the receiver.

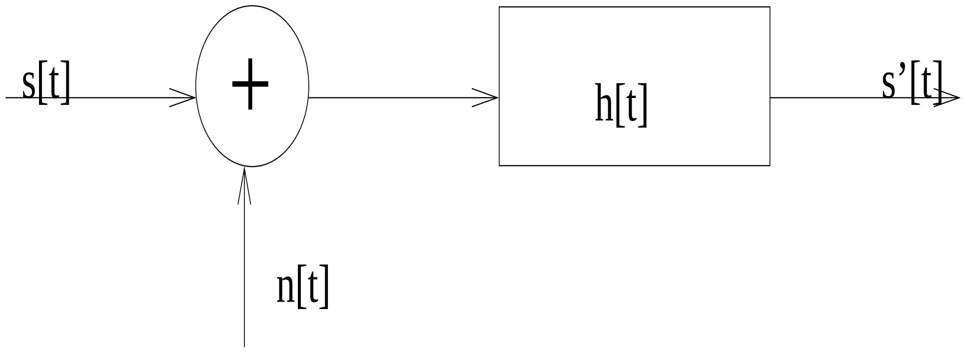

Figure 3.1: A noise model.

If we omit the Lombard effect, we are able to simplify our system to the one similar to the one shown in Figure 3.1. Hence, one can represent the corrupted signal as,

$$s'(t) = (s(t) + n(t)) * h(t), \tag{3.1}$$

where $n(t)$ and $h(t)$ are the overall additive noise and channel effect or channel noise of the environment respectively, $s'(t)$ and $s(t)$ are the corrupted speech signal and the clean speech signal respectively. The additive noise component is also regarded as uncorrelated to the speech signal.

In our research, we are interested in mismatched condition that are caused by noises, which corrupt only parts of the speech signal or **partial and temporal corruption** as termed by [42]. There is a substantial amount of research on speech recognition that deals with partial and temporal corruption. These can be categorized into two broad classes. In the first class, some examples of the acoustics in the operating environment can be collected and some portions can be extracted. We may regard this portion as pure noise. I.e., the $n(t)$ term in Equation 3.1. The clean speech model in training is then adjusted to match the noise characteristics. This class of techniques is called **model-based approach** by Gales in [19]. In the second class of technique, if the corruption in the operating environment is either time-limited or band-limited, the use of the clean speech model and omit those corrupted components (frames or bands) in the decision process of speech recognition are omitted. I.e., the recognition decision depends only on the clean part of the speech signal. The theory behind these techniques is called missing feature theory (MFT) [36].

In our research, we were interested in the techniques in the later class because model-based approach may cause mismatches between the clean portion of speech and the adjusted model. In Section 3.1, we discuss the some background of the model-based techniques. We describe the MFT techniques in Section 3.2. In Section 3.3 we define the meaning of short-time noise and review the use of the extended union model that applies the MFT to deal with short-time noises.

3.1 Model-Based Techniques

The aim of this class of techniques is to modify the acoustic model trained in laboratory conditions to match those of the operating conditions. Usually, some portion of the corrupted speech signal are extracted and are regarded as pure noise. I.e., the term $n(t)$ in Equation 3.1. The noise statistics are gathered from this noise signal and the acoustic model of the speech signal is adjusted using the noise statistic.

Hence, in this section, we describe techniques that can efficiently adjust the speech model for noisy environments. One natural idea is to apply the technique of speaker adaptation to adapt the noise characteristics for the speech model. This is the idea behind the noise adaptation described in Section 3.1.2, where the clean model is adapted to the noisy environment using a technique similar to that of speaker adaptation.

Then, in Section 3.1.3, we describe how the means and variances of a speech state and noise state if can be combined if it is assumed the noise is additive. In other words, given sufficient statistics of the clean speech and noise samples, the sufficient statistics of the corrupted speech signal can be efficiently calculated. This method which gives satisfactory results if the background noise is stationary is known as the parallel model combination by Gales in [19]. A similar technique can be used to combine noise and speech HMMs if the noise is assumed to be short-time stationary. This can be done by extending the search space of the standard Viterbi algorithm. This method is called signal noise decomposition (SND) and is described in Section 3.1.4.

3.1.1 Notations

We assume the same notation as we used in Section 2.3. The use of $O(t)$, $O(f)$, $O(\tau)$ and $O(\upsilon)$ mean that O is the function of time, frequency, binned frequency and quefrency respectively. The use of superscript ^{l} would mean the function is expressed in the log spectral domain.

Hence, the process of creating mel frequency cepstral coefficient (MFCC) for each frame can be expressed as,

$$O(t) \overset{\text{FFT}}{\rightarrow} O(f) \overset{\text{Binning}}{\rightarrow} O(\tau) \overset{\text{Log()}}{\rightarrow} O^l(\tau) \overset{\text{DCT}}{\rightarrow} O^l(\upsilon). \tag{3.2}$$

3.1.2 Noise Adaptation

The model adaptation technique [23] is a way of adapting the speech to the additive noise. The idea is similar to that of speaker adaptation technique such as the use of the Maximum Likelihood Linear Regression (MLLR), in which an affined transformation is estimated from the adaptation data. In the case of noise adaptation, we can assume that the effect of noise to all models can be characterized by the same shift of mean. This shift of mean is also known as **bias** in [53]. [1]

A special version of the MLLR is derived in [53] to estimate this bias. It is shown that this method can reduce the error rate especially of short utterances. However, as pointed out in [39], a large amount of adaptation data is necessary to learn the mismatched conditions. It is difficult to obtain the data and large computation is required even if the data is available.

3.1.3 Parallel Model Combination (PMC)

If it is assumed that the noise presented in the operating environment is additive in nature, the question is to adjust the clean speech model to match the noisy condition This question is addressed by Gales [19]. The most fundamental problem in this question can be stated as follows. If we assume the noise additive, i.e.,

$$s'(t) = s(t) + n(t) \tag{3.3}$$

where s', s and n are the corrupted speech signal, clean speech signal and noise signal respectively, and they are functioning in the time domain. If we assume $S'(\upsilon)$, $S(\upsilon)$ and $N(\upsilon)$ are Gaussian distributed in cepstral domain, what are the relationships between them?

[1] In the case of speaker adaptation, the biases of different means are usually different

Solving this problem will mean that the corrupted speech model's parameter based on the clean speech and noise models. Notice that their relations are non-linear because in the extraction process of MFCC, log operation has been applied to those binned frequency coefficients. Gales[19] suggested several solutions to this problem and he called them techniques of parallel model combination.

we now take a close look at this problem. If given the cepstral means of a clean speech state and a noise state, and if it is known that the noise is additive as in Equation 3.3, these cepstral means can easily be transformed to their log-spectral representation by computing the inverse discrete cosine transform (IDCT) of the cepstral means. I.e.,

$$S(\upsilon) \stackrel{IDCT}{\to} S^l(\tau). \tag{3.4}$$

However, as the noise is additive in the time-domain, it would also mean they are additive in the binned-frequency domain,

$$S'(\tau) = S(\tau) + N(\tau). \tag{3.5}$$

If we re-express Equation 3.5 in the log spectral domain representation of the signal, it would be,

$$\begin{aligned} e^{S'^l(\tau)} &= e^{S^l(\tau)} + e^{N^l(\tau)}, \quad \text{or} && (3.6) \\ S'^l(\tau) &= log(e^{S^l(\tau)} + e^{N^l(\tau)}), && (3.7) \\ &= log(S(\tau) + N(\tau)). && (3.8) \end{aligned}$$

and, thus, the mean of the corrupted speech model is the expectation of Equation 3.8 for all τ. . i.e.

$$\mu = E[log(S(\tau) + N(\tau))]. \tag{3.9}$$

The intrinsic difficulty in computing the mean of the corrupted signal, μ. In general, computing the mean based on Equation 3.9, requires computation of the multidimensional integration against τ. Even if we assume that the binned frequency components are independent for each component, integration is still required in order to calculate the mean. There is no close form solution for even such simple case.

Gales suggested several possible solutions to solve this problem in [19]. Two are,

1. **Numerical Integration**

   One can perform numerical integration using techniques such as Gauss-Hermite quardratures to solve Equation 3.9. It is possible to reduce the dimension of the integration by combining the terms in speech and noise Gaussian distribution. Still, this approach is very expensive because, for each state, a one-dimensional integration has to be performed for the mean and a two-dimensional integration has to be performed for the covariance matrix.

2. **Log Normal Approximation**

   The fact that the observation is normally distributed in the log spectral domain implies that it is log-normally distributed in the spectral domain. One can approximate the mean and the covariance of the corrupted signal in the log spectral domain as a log-normal distribution and derive expression of its mean and covariance using the one for the clean speech and noise models.

There are other ways to compute the parameters. For example, one can randomly draw samples from both the clean speech and noise models and evaluate the parameters of corrupted speech model using those random samples [23]. This is a kind of Monte Carlo method and is shown to be useful if the number of samples drawn is more than 100.

Using these techniques, one can update the speech model following the scheme outlined in Figure 3.2. First of all, the estimated parameters of the noise and the speech model are transformed to log-spectral domain using the inverse DCT. The parameters of the corrupted signal are then computed using one of the above techniques such as numerical integration. Then, the estimated parameter of the

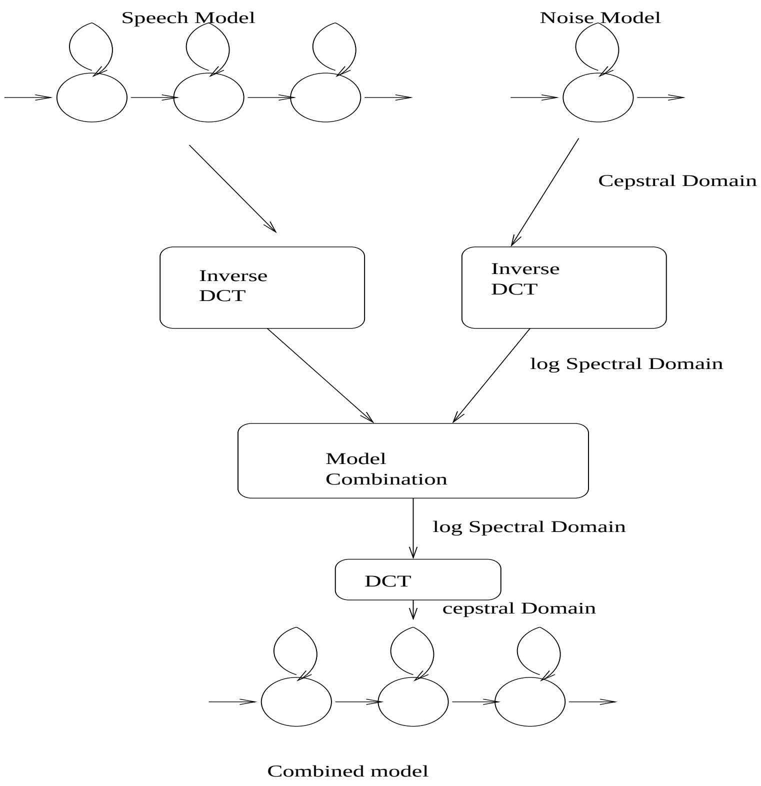

Figure 3.2: The parallel model combination.

corrupted speech model is transformed back to cepstral domain representation and the outcome is the final model.

Using the PMC means that extra computation is needed to update the model. However, as the corrupted speech model has the same topology as the clean speech models, there is no extra computation required in recognition.

Notice that in Figure 3.2, a single state is used to represent the noise signal. This effectively means that the noise signal is assumed to be stationary. In general, this assumption may not be true and the above method can be extended to short-time stationary signals or piece-wise stationary signals. This technique is described in the next section.

3.1.4 Signal Noise Decomposition

Signal noise decomposition can be regarded as a technique that modifies the standard Viterbi algorithm to concurrently decode noise and speech HMMs. If the noise signal is piecewise stationary and additive, one may express it as a HMM. In this case, the observation signal may be regarded as one generated from both speech and noise HMMs. Hence, if a similar paradigm as the maximum likelihood formulation, as we described in Chapter 2 is used. We need to enumerate all possible combinations of noise and speech state sequences. We can express the likelihood function, $\mathcal{L}$ of the observation in the log-spectral domain as,

$$\mathcal{L}(O_i^l(\tau)|Q_{speech}, Q_{noise}, \lambda_{speech}, \lambda_{noise}). \tag{3.10}$$

where Q_{speech}, Q_{noise} is the state sequence of speech and noise respectively.

Speech recognition in this situation can be regarded as a search of the maximum likelihood state sequence from all possible state sequences of speech and noise or,

$$\arg \max_{Q_{speech}, Q_{noise}} \mathcal{L}(O_i^l(\tau)|Q_{speech}, Q_{noise}, \lambda_{speech}, \lambda_{noise}).$$

The search can be done by introducing a 3-D Viterbi algorithm suggested by [63] as shown in Figure 3.3. This is called signal noise decomposition (SND). The calculation of the observation probability is done using methods such as the PMC described in last section.

The SND effectively combines a short-time stationary noise model with a speech model. However, the search space is increased significantly and this, thus, increases the amount of memory and computation required.

3.2 Missing Feature Theory (MFT) and Its Applications in Speech Recognition

Missing feature theory, also known as Missing data theory, is a branch of statistics that studies the inference and estimation of data sets in which some data missing.

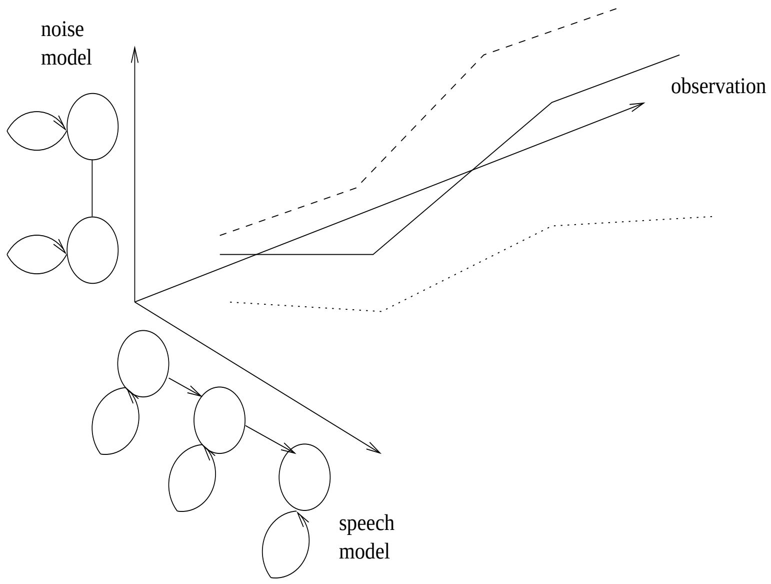

Figure 3.3: The 3-dimensional Viterbi algorithm.

Historically, it has been applied to subjects like survey analysis and multivariate analysis [38]. Unlike the model-based approach described in the last section (Section 3.1), missing feature theory does not model the corrupted speech signal. It takes a different approach in that the effect of those corrupted components in the speech signal are omitted.

The idea of ignoring unbelievable data in speech processing is supported by psycho-acoustics experiments. First, experiment shows that explicit spectro-temporal excisions typically do not cause decrease the intelligibility of the speech signal [61]. This means that the human auditory system is able to cope with speech with missing data. Second, the neural code exhibits a kind of behavior called the "capture effect" [44]. This means that those locally more intense components will dominate the neural response. Weaker signal components do not contribute to the neuronal output. Hence, corrupted components of the signal can be masked and can be regarded as the missing parts of the data.

MFT provides a solid foundation for dealing with noise with a weak assumption about noise and has been applied in speech segregation and vision [5].

To apply missing feature theory in speech recognition requires solutions to two problems.

1. The choice of components in the algorithm is very important. Some researchers have noted this problem. This is most remarkably reflected in [27] by Veth *et. al.*, "Missing Feature Theory in ASR: Make Sure You Miss the Right Type of Features". The title suggests that knowledge of what type of noise is very important in applying the missing feature theory in speech recognition. This is because the feature vector will follow a sequence of transforms in the feature extraction process and the corruption of one spectral component may affect all the MFCC coefficients. Hence, it can be seen that techniques in noisy speech recognition applying MFT always have a very clear assumption of the noise. For example, [42] assumes noise appeared in the time-domain and [43] suggests that noise may only corrupt certain sub-band of speech.

2. Another important problem is to detect the corrupted region of the signal. There are several difficulties in regard to this problem. First of all, detecting the corrupted signal is hard. Hence, the recognition result will be subjected to errors of the detector as well. Also, as speech recognition requires a sequence of transforms which easily make noise information appears in every component of a feature vector, the false rejection rate in this case may increase a lot. Finally, we noted that the use of signal detection can always result in a separate stage in the recognition process. Thus, the computation of the recognition may increase,

We now describe several examples of the application of MFT in speech recognition.

In [37] Lipmann and Carlson suggest that the cepstral domain components can be omitted from the likelihood computation if they are known to be corrupted.

The calculation of the likelihood of the Gaussian distribution of a state in a HMM can be written as,

$$b(o) = \sum_{k=1}^{M} w_k \prod_{l=1}^{D} N(o_l|\mu_{kl}, \sigma_{kl}^2), \tag{3.11}$$

where M is the number of mixtures and D is the number of components of the cepstral vector and we have assumed the covariance matrix is diagonal.

If some cepstral components are known to be corrupted, then one can rewrite Equation 3.11 as

$$b(o) = \sum_{k=1}^{M} w_k \underbrace{\prod N(o_l|\mu_{kl}, \sigma_{kl}^2)}_{Corrupted} \underbrace{\prod N(o_l|\mu_{kl}, \sigma_{kl}^2)}_{Uncorrupted} \tag{3.12}$$

Missing feature theory suggests that one can omit the terms related to corrupted data in Equation 3.12. That is,

$$b(o) = \sum_{k=1}^{M} w_k \underbrace{\prod N(o_l \mu_{kl}, \sigma_{kl}^2)}_{Uncorrupted} \tag{3.13}$$

Another interpretation of Equation 3.12 is that the likelihood is calculated by integrating the original likelihood function in Equation 3.12 over the corrupted features and thus all terms related to missing features are eliminated.

Lipmann and Carlson have tested this technique using time-invariant (high pass and low pass noise) and time-varying noise (systematic excision of some of the spectro-temporal component). They provided the recognizer the prior knowledge of where the cepstral component was corrupted and applied this knowledge in the modified likelihood computation in recognition. This simple technique represents a surprisingly great improvement (relatively 90 % gain) in a corpus with artificial corruption in the filter-bins. Another strength of this technique is that only simple modification has to be made to existing speech recognition software. This makes the developing time much shorter.

The result is highly encouraging. However, we noted from their paper that the position of the corrupted cepstra was actually known. Therefore, it can be expected the result would be poorer if the detection of noisy cepstra need to be done. Similar techniques have also been used in [11] for word spotting.

While in Lipmann's work, the corrupted signal is omitted in the decision process, other researchers suggest that one can compute the corrupted data values from those uncorrupted ones, or imputation. The imputed value is useful if the reconstruction of the data is necessary. In [13], Cooke *et. al.* suggested that there are several ways of achieving the reconstruction. The simplest way is to replace the value of the corrupted components by the mean of other uncorrupted components. A better approach is to use the knowledge of the reliable components in conjunction with the covariance. Yet another way is to draw random samples from the distribution of random data and treat these as the value of the reliable data.

In regard to detecting the position of the corruption, some people have tried to find the positions of corruption using spectral subtraction. In [32], Kermorvant and Morris showed that missing data imputation have a performance similar to that of spectral subtraction in additive noise. However, the two techniques are shown to be additive in [13].

Missing feature theory can be applied not only in the spectral and the cepstral domain. It can also be extended to other situations such as multi-band systems[43]. In the next section, we discuss techniques in which it is assumed that the corruptions are limited in the time domain.

3.3 Extended Union Model (EUM) and Techniques That Deal with Partial and Temporal Corruption

In previous sections, we discussed some general techniques that deal with mismatched conditions, which are caused by additive noise. In this section we narrow our discussion to a technique that makes use of missing feature theory to deal with mismatched conditions caused by short-time noise. It is called the extended union model. Before we outline this technique in Sections 3.3.3- 3.3.4, we first discuss the precise definition of short-time noise or partial and temporal corruption (in Section 3.3.1) and general techniques for dealing with them (in Section 3.3.2).

3.3.1 Definition of Partial and Temporal Corruption

In this section, we define the meaning of partial and temporal corruption is similar to that of impulse noise. The best definition of impulse noise available may be that "Impulse noise is noise consisting of random occurrences of energy spikes having random amplitude and spectral content" [20]

This definition implies that there are three basic properties of impulse noise.

1. Random occurrences

   This means impulse noise occurred unexpectedly.

2. Random amplitude

3. Random spectral content

   This means learning is useless for impulse noise.

In our work, we further assumed that we are dealing with impulse noise which was

4. Shorter than the speech signal and only affected a portion of the speech signal. However, the duration is random under this constraint.

If we do not assume that noise is shorter than the speech, the impulse noise we are dealing with is more intuitively to be characterized as general additive noise because the whole speech signal is actually corrupted. We called those noises that satisfy the above four conditions as **partial and temporal corruption**. This term was first used in [42].

3.3.2 General Approaches in Dealing with Partial and Temporal Corruption

There are many existing techniques to handle partial and temporal corruption. The first way is to apply techniques such as the model-based techniques (described in Section 3.1). In particular, those techniques which assume a mismatched condition work very well For example, partial temporal corruption can be regarded as a kind of piecewise stationary noise and using techniques such as PMC and SND to adjust the clean speech model statistics. However, missing feature theory (described in last section) provides yet another attractive alternative compared to the model-based approach. We notice that model-based techniques actually assume some prior knowledge of the noise. For instance, in PMC or SND, we assume the knowledge of the form of the stochastic model of the noise. In missing feature theory (MFT)-based technique, we do not make such an assumption. MFT-based technique potentially would then be able to handle unknown noises.

3.3.3 Theory of Extended Union Model

Extending the idea of extracting information from only clean frames, J. Ming *et al.* proposed an idea in [42] called the extended union model (EUM). We described the idea in this section.

The theory of the general union model and the extended union model can be found in [42]. As only the extended union model (EUM) is used in ASR, we describe it in more detail in this section. The formulation of the theory can be stated as follows. Suppose one assumes that among T frames, $O = (o_1, \ldots, o_T)$, in which M of them are corrupted. However, we do not know which M frames can be believable. From a basic combinatoric argument, we know that there are C_M^T possible patterns of corruptions and we denote them $l_1, \ldots, l_{c_M^T}$. For each possible pattern of corruption l_i, frames $(o_{i_1}, \ldots, o_{i_{T-M}})$ are not corrupted. We can express the probability of corruption for this pattern l_i as,

$$P(O) = P(o_{i_1} \wedge \ldots \wedge o_{i_{T-M}}). \tag{3.14}$$

In Equation 3.14, we have omitted the contributions of the corrupted frames as suggested by missing feature theory [36].

We want to extend Equation 3.14 to model the event, in which *any* of the C_M^T patterns can occur. One can express this uncertainty using the union operation. Define

$$\mathring{O}_{\vee(T)}^{M} = \bigvee_{i=0}^{C_M^T} l_i, \tag{3.15}$$

$$= \bigvee_{i=0}^{C_M^T} (o_{i_1} \wedge o_{i_2} \wedge \ldots o_{i_{T-M}}). \tag{3.16}$$

We called $\mathring{O}^M_{\vee(T)}$ the extended union model (EUM) of order M in T frames. It represents the event of *any* occurrence of M-corrupted frames among T events.

Here is one example of EUM with four frames. Applying Equation 3.16. Table 3.1 shows the EUM of orders $0, 1, 2, 3$.

M	Extended Union Model
0	$o_1 o_2 o_3 o_4$
1	$o_1 o_2 o_3 \vee o_1 o_2 o_4 \vee o_1 o_3 o_4 \vee o_2 o_3 o_4$
2	$o_1 o_2 \vee o_1 o_3 \vee o_1 o_4 \vee o_2 o_3 \vee o_2 o_4 \vee o_3 o_4$
3	$o_1 \vee o_2 \vee o_3 \vee o_4$

Table 3.1: Examples of EUM of different orders in the case of 4 frames with orders $0, 1, 2, 3$. They respectively represent that $0, 1, 2, 3$ frames is/are missing.

Applying Equation A.2 described in Appendix A and assuming independence of events, we can compute $P(\mathring{O}^M_{\vee(T)})$ in Equation 3.16 as

$$P(\mathring{O}^M_{\vee(T)}) \approx \sum_{i=0}^{C_M^T} P(o_{i_1}) P(o_{i_2}) \ldots P(o_{i_{T-M}}). \tag{3.17}$$

It is intuitive to define $P(\mathring{O}^M_{\vee(T)}) = 0$ when $M = 0$ or $T = 0$ or $M > T$ because in all these cases , $\mathring{O}^M_{\vee(T)} = \emptyset$.

Equation 3.17 suggests that the probability of the EUM has a nice property; The corrupted frames will only make a small contribution to it and clean frames dominate the probability of EUM.

Direct computation of the probability of the EUM defined in Equation 3.17 requires $C_M^T * (T - M)$ multiplications and C_M^T additions. If T is large, a large number of multiplications is required. Table 3.2 shows the typical number of multiplications required for the computation of $P(\mathring{O}^M_{\vee(T)})$ when $T = 100$.

The EUM with the order $1, 2, 5, 10$ respectively represents 1%, 2%, 5%, 10% corruptions in 100 frames. This is even true for 5% corruptions. The number of multiplications is large so that this can be regarded as an intractable task.

T	M	Multiplications
100	1	9900
100	2	485100
100	5	7152314400
100	10	1.55×10^{15}

Table 3.2: Number of multiplications required in computation of $P(\mathring{O}^{M}_{\vee(T)})$.

3.3.4 Application of the EUM in Speech Recognition Against Short-Time Noise

Extend union models (EUM) are useful in handling corrupted frames in the speech signal. In [42], Ming and Smith proposed one possible use of EUM to speech recognition against partially temporal corruption.

In their paper, they introduced the notation of a segment which consists of constant number of consecutive frames. They partitioned an utterance into consecutive segments $(z_1, \ldots, z_N)$. They applied EUM at the segment-level and they tried to find the best state sequence $\hat{Q}_1^T$ which could optimize the segment-based extended union probability.

$$\hat{Q}_1^T = \arg\max_{Q} P(\bigvee_{i=0}^{C_M^N}(z_{i_1} \wedge z_{i_2} \wedge \ldots z_{z_{T-M}})|Q)$$

where they omitted the conjunction operator in the above expression. Within each segment, the probability of frames were combined together as an joint event, i.e., $P(z_N|Q) = \prod_{i=1}^{S} P(o_i|Q)$ where S is the number of frames in the segment. Since the number of segments is much less than the number of frames in an utterance, the computation of EUM can be significantly reduced. However, the assumption that corruptions occurs in segment-level implies that even if part of a segment is corrupted, the whole segment is regarded as corrupted. This will result in loss of useful information for decision.

Though Ming and Smith proposed a way to solve the problem of computation of EUM give the state sequence, their model cannot be easily applied for every state sequence. Therefore, they have chosen to apply EUM on the N-best outcomes only. This implies that corrupted frames may still be allowed to affect the N-best generation process.

In this thesis, we will describe an algorithm called the frame-skipping Viterbi algorithm that assume the corruption in the frame-level. At the same time, the algorithm use computation similar to the standard Viterbi algorithm. Our proposed algorithm has some interesting relation with the EUM. We will describe this in Chapter 4.

3.4 Summary

In this chapter, we reviewed two classes of techniques for noisy speech recognition. We categorize techniques into two classes. One is the model-based technique which adjusts the speech model so that it is more adaptive to noise. Noise adaptation, parallel model combination and signal noise decomposition belongs to this class. The second class are those based on missing feature theory such as the marginalization method described by Lipmann and imputation described by Cooke. The extended union model, which extracts information from the unaffected part of speech also belongs to this class. Our idea can be regarded as another attempt to use missing feature theory. We end this chapter by summarizing the techniques described in Figure 3.4.

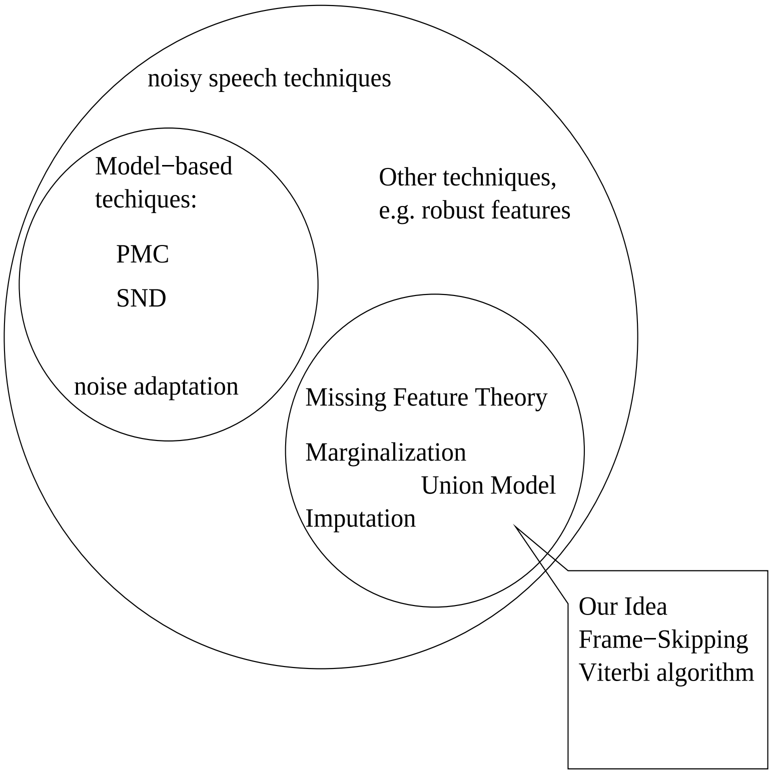

Figure 3.4: A classification of noisy speech techniques described in Chapter 3.

Chapter 4

Theory of the Frame-skipping Viterbi Algorithm (FSVA)

In Chapter 1, we mentioned our main goal in this thesis is to develop algorithms which can handle partial and temporal corruption. We explained what we mean by partial temporal corruption in Section 3.3.1. In short, what we mean by short-time noise has the following four properties, 1) random occurrence, 2) random amplitude, 3) random spectral content and 4) shorter than the speech signal such that only portions of the speech signal are affected. Perhaps the last property is the most important one because it motivates us to use the approach that extracts useful information from the unaffected part of the speech signal.

There are other attempts that extracts information from the unaffected portion of speech. One proposal described in Section 3.3 is the extended union model (EUM) proposed by Ming and Smith in [42]. We discussed their use of segment-based extended union model to deal with partial and temporal corruption (PTC). We noticed that their approach assumes corruption occurred in segment-level. This assumption may not be reasonable if the noise appears in a couple of frames within a segment. Also, as they re-scored N-best results generated from the standard Viterbi algorithm. The resulting performance is still easily affected by PTC.

In this thesis, we take a different approach. We developed a dynamic algorithm, called frame-skipping Viterbi algorithm (FSVA), which omits the K worst performing frames in the Viterbi search. Since the FSVA assumes frame-level corruption, it will not suffer from the same problems as that of EUM. Moreover, our algorithm is shown to have a similar computation requirement as the standard Viterbi algorithm.

In this chapter, we describe the theory and the implementation of the FSVA. We describe its motivation in Section 4.1. We observe that the deficiency of the Viterbi algorithm in PTC stemmed from the computation of likelihood. Then, we propose an improvement to the likelihood function by reformulating a robust version of likelihood that omits the contribution of K worst performing frames, called the **robust likelihood**, in Section 4.2. The problem of speech recognition in PTC thus becomes the problem of finding a state sequence that optimizes the robust likelihood. We develop the dynamic algorithm, the FSVA and describe it with pseudo-code in Section 4.3. The FSVA can be implemented in two ways, one by changing the state-space and one by changing the topologies of the HMMs. These implementations are described in Section 4.4.

There are many interesting relations between the FSVA and other noisy speech techniques. We discuss the relation between the FSVA and the signal noise decomposition (SND) and the relation between the robust likelihood and the extended union model. These are discussed in Section 4.5. We conclude and summarize this chapter in Section 4.6.

4.1 Motivation : Deficiency of Maximum Likelihood as a Criterion Function in Partial and Temporal Corruptions

As we mentioned in Section 2.4, a HMM-based ASR system searches for the state sequence that best explains the observation sequence. Mathematically,

$$\arg\max_{Q_1^T} L(\, Q_1^T,\; O_1^T) \;\;=\;\; \arg\max_{Q_1^T} \log P(\, O_1^T,\; Q_1^T)$$

Here Q_1^T is a state sequence and O_1^T is the observation sequence.

The above expression is basically equivalent to

$$Q = \arg\max_{Q \in Q^*} (\underbrace{\log \pi_{q_1}}_{\text{init. prob.}} + \underbrace{\sum_{t=1}^{T-1} \log a_{q_t q_{t+1}}}_{\text{trans. prob.}} + \underbrace{\sum_{t=1}^{T} \log b_{q_t}(o_t)}_{\text{obs. prob.}}), \tag{4.1}$$

where $Q*$ is the set of all possible state sequences.

Similar to using the average as the mean of the samples, using Equation 4.1 to decide the best state sequence is easily affected by outlier frames. The effect of outlier frames is especially large to the observation terms because numerically the observation probability usually dominates Equation 4.1.

4.2 The Robust State Sequence

In Section 4.1, we notice that the standard Viterbi algorithm is not robust against outlier frames or corrupted frames. This is because outlier frames can easily affect the values of the observation term in the likelihood function. This fact motivates us to derive a new criterion function that is not easily affected by corrupted frames. In statistics, one use the trimmed mean, which ignores $x\%$ of sample data, instead of the average as the samples' mean. Analogous to the use of trimmed mean, we propose a likelihood function that omits the K worst-performing frames for a particular state sequence. We can express this idea more formally as follows.

For a given state sequence, Q_1^T, if we re-index the frames in O_1^T into $(o_{t_1}, \ldots, o_{t_T})$, such that

$$p(o_{t_i}|q_{t_i}) \leq p(o_{t_{i+1}}|q_{t_{i+1}}).$$

We can define a new criterion function for a state sequence Q_1^T to replace the observation likelihood,

$$\begin{aligned} \phi_K(O_1^T|Q_1^T) &= \sum_{i=K+1}^{T} \log p(o_{t_i}|q_{t_i}), \\ &= \sum_{t \notin \{t_1,\ldots,t_K\}} \log p(o_t|q_t). \end{aligned} \tag{4.2}$$

[1]Since we have omitted the values of the worst performing K frames in a given state sequence in Equation 4.2, $\phi_K(\,O_1^T|\,Q_1^T)$ is more robust than observation likelihood term in Equation 4.1.

We called $\phi_K(\,O_1^T|\,Q_1^T)$ the **robust likelihood function**.

$\phi_K(\,O_1^T|\,Q_1^T)$ is usually[2] an increasing function of K for a give path Q_1^T because

$$\begin{aligned}\phi_K(\,O_1^T|Q) &= \phi_{K+1}(\,O_1^T|Q) + \log p(o_{t_{K+1}}|q_{t_{K+1}}),\\ &\le \phi_{K+1}(\,O_1^T|Q).\end{aligned} \tag{4.3}$$

Another way to understand the robust likelihood can be followed from a combinatoric point of view. Suppose we know that K frames are corrupted in a T-frame state sequence. However, we do not know the exact position of the corruptions. Similar to the discussion in EUM in Section 3.3.3, one knows that there are C_K^T possible patterns of corruption $l_1, \ldots, l_{c_K^T}$. For each possible pattern of corruption l_i, frames $(o_{i_1}, \ldots, o_{i_{T-K}})$ are not corrupted. Robust likelihood in Equation 4.2 can be rewritten as

$$\begin{aligned}\phi_K(\,O_1^T|\,Q_1^T) &= \max_{i=0}^{C_K^T} p(l_i|\,Q_1^T),\\ &= \max_{i=0}^{C_K^T}(p(o_{i_1}|q_{i_1})\ldots p(o_{i_{T-K}}|q_{i_{T-K}})).\end{aligned} \tag{4.4}$$

$p(l_i|\,Q_1^T) = p(o_{i_1}|q_{i_1})\ldots p(o_{i_{T-K}}|q_{i_{T-K}})$ follows from the assumption of independence of frames and the fact that the corrupted frames can be omitted. Equation 4.4 effectively means we are finding the most likely patterns from the all possible patterns of K-frame corruption.

Both Equation 4.2 and Equation 4.4 define the robust likelihood and they are equivalent.

The above discussion suggests that one can obtain better performance for speech recognition in PTC by optimizing the robust likelihood function, instead

[1]The terms of initial probability and transition probabilities are omitted for simplicity.

[2]In speech recognition, it is usually true to assume $p(o_t|q_t) < 1$ for all t. Though strictly speaking $p(o_t|q_t)$ is a continuous likelihood distribution and thus can be larger than 1, the assumption still usually holds because of the high dimensionality and sparseness of the space.

of the likelihood function. Hence, one can reformulate the problem of speech recognition in PTC as the search of the state sequence with maximum robust likelihood. I.e.,

$$\hat{Q}_1^T(K) = \arg\max_{Q_1^T} \left(\phi_K(O_1^T|Q_1^T) + \sum_{t=1}^{T} \log p(q_t|q_{t-1}) \right). \tag{4.5}$$

We call the state sequence with the maximum robust likelihood, $\hat{Q}_1^T$ as the **maximally robust state sequence**.

Notice that in Equation 4.5, we only omit the contributions of the outlying observations but retain the state transitions such that the length of state sequence remains T. This preserves the duration information.

Direct computation of Equation 4.5 requires resolving the problem of sorting the state sequence in Equation 4.2 which has $O(T \log T)$ computation. Also, the robust likelihood is applied to every possible state sequence for an observation sequence. Hence, the computation is very expensive and an efficient algorithm is necessary. In the next section, we describe one solution , the frame-skipping Viterbi algorithm (FSVA) to tackle this computation problem.

4.3 The Frame-skipping Viterbi Algorithm (FSVA)

In the last section (Section 4.2), we describe a new criterion function, called the robust likelihood, that can omit the effect of outlier frames. The question becomes how we can efficiently search for the maximally robust state sequence. This is an important question because direct computation of the robust likelihood function is intractable for any reasonable values of T. In this section, we try to solve this computation problem by establishing a recursive relationship of robust likelihood function. This recursive relation allows us to search for the maximally robust state sequence in the similar manner as the standard Viterbi algorithm.

We begin from the definition of robust state sequence at time t as in Equation 4.4,

$$\phi_K(O_1^t|Q_1^t) \quad = \quad \max_{i=0}^{C_K^t} p(l_i|Q_1^t). \tag{4.6}$$

For a fixed value of K, we can divide the C_K^t patterns of corruption $(l_1, \ldots, l_{C_K^t})$ into two subsets. In the first set, L_1, o_t is omitted. The second set, L_2, does not.

Hence, by rearranging the terms inside the max operator. Equation 4.6 can be rewritten as

$$\phi_K(\ O_1^t|\ Q_1^t) = \max(\max_{l_i \in L_1} p(l_i|\ Q_1^t), \max_{l_i \in L_2} p(l_i|\ Q_1^t))$$

For L_1, since o_t has been omitted, there are only $K-1$ omissions occurred in the previous $t-1$ frames. Thus, one can express $\max_{l_i \in L_1} p(l_i|\ Q_1^t)$ as $\phi_{K-1}(\ O_1^{t-1}|\ Q_1^{t-1})$.

For L_2, since o_t has not been omitted, there must be K corruptions occurred in the previous $t-1$ frames. Thus one can express $\max_{l_i \in L_2} p(l_i|\ Q_1^t)$ as $\phi_K(\ O_1^{t-1}|\ Q_1^{t-1}) b_j(o_t)$.

Hence, the above discussion suggests that Equation 4.7 can be rewritten as,

$$\phi_K(\ O_1^t|\ Q_1^t) = \max(\phi_{K-1}(\ O_1^{t-1}|\ Q_1^{t-1}), \phi_K(\ O_1^{t-1}|\ Q_1^{t-1}) b_{q_t}(o_t)), \tag{4.7}$$

Notice that we can verify the above arguments by checking the cardinality of L_1, L_2 and $L_1 \vee L_2$.

$$\begin{aligned} & |L_1| + |L_2| \\ = \ & C_K^{t-1} + C_{K-1}^{t-1}, && (4.8) \\ = \ & C_K^t, && (4.9) \\ = \ & |L_1 \vee L_2|. \end{aligned}$$

Equation 4.9 is obtained from Equation 4.8 using the well-known Pascal's formula [8].

Equation 4.7 suggests that if we want to obtain $\phi_K(\ O_1^t|\ Q_1^t)$, we will need to obtain $\phi_{K-1}(\ O_1^{t-1}|\ Q_1^{t-1})$ and $\phi_K(\ O_1^{t-1}|\ Q_1^{t-1})$. To obtain $\phi_{K-1}(\ O_1^{t-1}|\ Q_1^{t-1})$, we need to recursively obtain $\phi_{K-2}(\ O_1^{t-2}|\ Q_1^{t-2})$ and $\phi_{K-1}(\ O_1^{t-2}|\ Q_1^{t-2})$. In general, to obtain $\phi_K(\ O_1^t|\ Q_1^t)$, we need to compute $\phi_k(\ O_1^\tau|\ Q_1^\tau)$ for all k with $0 < k < K$ and τ with $t - K < \tau < t$. This basically suggest that search of the best $\phi_K(\ O_1^t|\ Q_1^t)$ have an extra dimension compared to the standard Viterbi algorithm.

As we discussed in the last section, speech recognition in PTC may be reformulated as searching the maximally robust state sequence. The best robust path, $\hat{Q}_1^T$ is given by searching the state sequence with best robust likelihood,

$$\hat{Q}_1^T(K) = \arg\max_{Q_1^T} \left(\phi_K(O_1^T | Q_1^T) + \sum_{t=1}^{T} \log p(q_t|q_{t-1}) \right). \tag{4.10}$$

The recursive relation we showed in Equation 4.7 suggests that Equation 4.10 can also be expressed recursively. Lets define a robust version of δ function in state j, with skip k at time t (The standard version can be found in Equation 2.19.) as,

$$\delta_t(j,k) = \arg\max_{Q_1^t} \left(\phi_k(O_1^t | Q_1^{t-1}, q_t = j) + \sum_{t=1}^{t} \log p(q_t|q_{t-1}) \right). \tag{4.11}$$

Similar to the proof of recursive relation in Equation 4.7, Equation 4.11 can be written recursively as,

$$\delta_t(j,k) = \max_i \left[a_{ij} \max(\delta_{t-1}(i, k-1), \delta_{t-1}(i,k) b_j(o_t)) \right] \quad \text{for } 0 < k < K. \tag{4.12}$$

In the standard Viterbi algorithm, we use backtracking to find the best state sequence. In the FSVA, we use similar idea to search for the maximally robust state sequence. In every state j with every skip k, we augment yet another two pieces of information. The first one, ψ, stores the previous state index that gives maximum robust likelihood at state j with skip k. This is similar to the treatment in the standard Viterbi algorithm. The second one, φ, stores the previous skip index that gives the maximum robust likelihood at state j with skip k. From Equation 4.12, we notice that this skip index can only be either k or $k-1$. The two pieces of information allow us to trace-back the maximally robust state sequence in a similar manner as the backtracking step in the standard Viterbi algorithm.

We show that the FSVA for skipping K frames for an N-state HMM in the next page. The observation sequence is assumed to have length T.

If we denote the ending node of the network of HMMs as V or if we look at the partial path at time T at node V, $\delta_T(V,k)$, we actually compute all the best robust paths $\hat{Q}_1^T(1), \ldots, \hat{Q}_1^T(K)$ and robust likelihoods $\phi_1(\ O_1^T | \hat{Q}_1^T(K)), \ldots, \phi_K(\ O_1^T | \hat{Q}_1^T(K))$

for skipping 0 to K frames. Hence, if we skip a fixed number of frames, K, we will get $K+1$ solutions corresponding to skips 0 to K (One can also see that in the backtracking stage of the algorithm.) This property is used to automatically determine the number of skips for FSVA in later chapters.

1. **Initialization**

   for $1 \leq i \leq N$, $\delta_1(i,0) = \pi_i, \psi_1(i,0) = 0, \varphi_1(i,0) = -1$,
   for $1 \leq i \leq N$, $1 \leq k \leq K$, $\delta_1(i,k) = 0$.

2. **Update of Viterbi Score** for t from 1 to $T-1$, calculate $\delta_{t+1}(i,k)$ using Equation 4.12.

$$
\begin{aligned}
\text{Initialization,} \quad & \text{for } 1 \leq j \leq N, \\
\delta_{t+1}(j,0) &= \max_i a_{ij}\delta_t(i,0)b_j(o_t)), \\
\psi_{t+1}(j,0) &= \arg\max_i a_{ij}\delta_t(i,0)b_j(o_t)), \\
\varphi_{t+1}(j,0) &= 0. \\
\text{Update,} \quad & \text{for } 1 \leq j \leq N, 1 \leq k \leq K, \\
\delta_{t+1}(j,k) &= \max_i \left[a_{ij} \max(\delta_t(i,k-1), \delta_t(i,k)b_j(o_t))\right], \\
\psi_{t+1}(j,k) &= \arg\max_i \left[a_{ij} \max(\delta_t(i,k-1), \delta_t(i,k)b_j(o_t))\right], \\
\varphi_{t+1}(j,k) &= \begin{cases} k-1 & \text{if } \delta_t(i,k-1) > \delta_t(i,k)b_j(o_t) \\ k & \text{otherwise.} \end{cases}
\end{aligned}
$$

3. **Termination** For k in 0 to K,

$$\phi_k(\ O_1^T|Q_1^T) = \max_{j=1}^{N} \delta_T(j,k) \quad \text{for } 0 \leq k \leq K.$$

4. **Back-Tracking** For k in 0 to K, The robust state sequence that skips k frames can be found by,

$$\tilde{q}_t = \psi(\tilde{q}_{t+1}, \tilde{m}_{t+1}) \quad \text{for } 1 \leq i \leq T-1,$$
$$\tilde{m}_t = \varphi(\tilde{q}_{t+1}, \tilde{m}_{t+1})) \quad \text{for } 1 \leq i \leq T-1.$$

   where m_t and q_t respectively denotes the skip and state indices at time t.

4.4 Implementations of FSVA

In this section, we describe two implementations of FSVA: 1) state-space extension and, 2) topological-space extension. The first approach aims to implement Equation 4.12 by extending the state-space in the Viterbi algorithm. The second approach aims to implement the proposed algorithm by changing the HMMs' topologies. This will allow the FSVA to be easily implemented using standard speech recognition software (such as HTK).

4.4.1 State-Space Extension Approach

Similar to the exact N-best algorithm proposed in [58], one can estimate the best K-frame skipped state sequence by augmenting the information stored for each state. Figure 4.1 shows a two-state left-to-right HMM for skipping 1 frame. The HMM has two states, state 1 and state 2. The (j_k)-th box stores the maximally robust likelihood of state j with skip k. For example, the (1_1)-th box stores the partial score at state 1 with 1 skip. Each box represents the extended state in which we stores $K + 1$ scores where the k-th score is the current best score given that k frames have been omitted/skipped in its history. The original Viterbi partial path score is stored as the 0-th score. During the update, in addition to comparing the N possible incoming arcs, we also need to decide whether this frame should be skipped. Under this modification, the computation is at most $K + 1$ times of the original Viterbi search. For the storage, it is $K + 1$ times larger than the Viterbi search. However, in reality, the computation is only slightly bigger than the original Viterbi because $b_j(o_t)$ needs to be computed only once per state which typically dominates the computation. This idea can be readily turned to apply in continuous speech recognition.

4.4.2 Topological-Space Extension Approach

Suppose each word is represented by a left-to-right HMM with 3 states and no skipping arcs. To extend this HMM to skip one frame, one can extend the

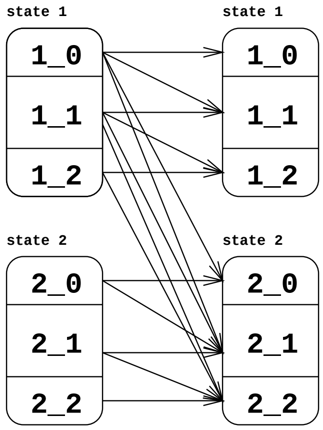

Figure 4.1: Search rules in the FSVA.

HMM's topology to include a set of *skip-states* denoted as ($1_s, 2_s$, and 3_s) and a set of *skipped states* ($1_m, 2_m$, and 3_m) as shown in Figure 4.2. The *skip-states* are visited whenever a frame is skipped. Ideally, we would like the skip-states to return an observation probability of 1 resulting in no contribution to the total log likelihood. Instead, it can be approximated by a uniform observation distribution that returns a constant log likelihood of c. Since all paths that we consider include the same number of skips, the introduction of c causes a constant increase in the path likelihood but with no effect in selecting the best path. Notice that there is no self-loop at the skip-states. Instead, it is forced to proceed to one of the *skipped states* $1\_m, 2\_m, 3\_m$. The skipped states denote paths that have already skipped one frame. Because of this, a skipped state can transit only to other skipped state. The skipped states' statistical properties, such as observation distributions and transition probabilities, are duplicates of the original states. We refer this expanded model as frame-skipping HMM (FSHMM).

The FSHMM can be generalized to skip more frames. For an N-state HMM, each additional skip increases the total number of states by 2N. Because we are adding $2N$ states, the FSHMM is not as memory-efficient as the state-space extension approach.

FSHMM provides us an alternative view for obtaining the maximally robust state sequence. This also allows us to extend the frame-skipping Viterbi algorithm. For example, the skip state can be modified to have an observation distribution that captures transmission noises or corruptions. Instead of skipping a fixed number of frames, skipping probabilities can be added on the transitions between the skipped-states.

It is possible to extend the above approach to continuous speech recognition. However, the extension is quite tedious and we will discuss its difficulty and implementations in Appendix C. In general, this requires exponential amount of memory and thus is very hard to be put in practice.

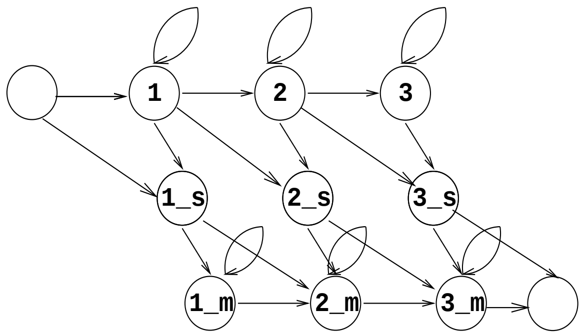

Figure 4.2: The topology of an frame-skipping HMM.

4.5 Relations between the FSVA and Other Noisy Speech Recognition Techniques

We discussed the theory and implementations of the FSVA in previous sections and found that the FSVA is useful for dealing with partial and temporal corruptions (PTC). It is interesting to understand the underlying relations between the FSVA and other noisy speech recognition techniques.

In this section, we describe the relations of FSVA and other techniques. We observe from the topologies of an FSHMM that it is basically a 2-dimensional structure. This suggests the potential relation with signal noise decomposition (SND) described in Section 3.1.4. We discussed this topic in Section 4.5.1. It turns out that FSVA can be interpreted as a combination of the speech HMM and a Markov model that models the frame skipping characteristic of the channel.

We also mentioned another technique that deals with PTC in Section 3.3.

This is called the extended union model (EUM) in [42]. This technique and the FSVA have different origins. Previously, Ming *et. al.* [42] applied it in segment-level. We have argued in Section 3.3 that the segment-based EUM has drawbacks. In Section 4.5.2, we formulate a **frame-based EUM search**. This search basically assumes the corruption occurred in the frame-level instead of the segment-level.

We observe that the formulation of the frame-based EUM search has many similarities with that of the search of the robust likelihood search. This observation leads us to consider two questions. First, if their formulations are very similar, it is natural to apply similar search technique of the FSVA can be applied in the frame-based EUM search. In Section 4.5.3, we describe one such idea, the **frame-based EUM Viterbi algorithm**, which can efficiently search the state sequence with maximum probability of EUM. Second, we like to understand the relation between the search of *robust likelihood* and *probability of EUM*. We find that in practice, the two criteria are approximately the same. However, search of robust state sequence has an extra advantage in practice. This interesting fact is discussed in Section 4.5.4.

4.5.1 Signal Noise Decomposition Interpretation of the FSVA

If we look at the topology of an FSHMM, we notice that it is basically a 2-dimensional HMM in which one dimension describes the state in speech HMM and the other describes the number of skips. Thus, it is natural to "decompose" the FSHMM into two Markov models, one is a speech HMM and one is a Markov model that indicates how many frames are skipped. This is illustrated in Figure 4.3. The decomposition allows us to analyze our current method by examining the skip Markov model only.

It is interesting to explore more about the skip Markov models. Figure 4.4 shows the skip Markov model with infinite number of skips. We observe that there are basically two types of state in this Markov model.

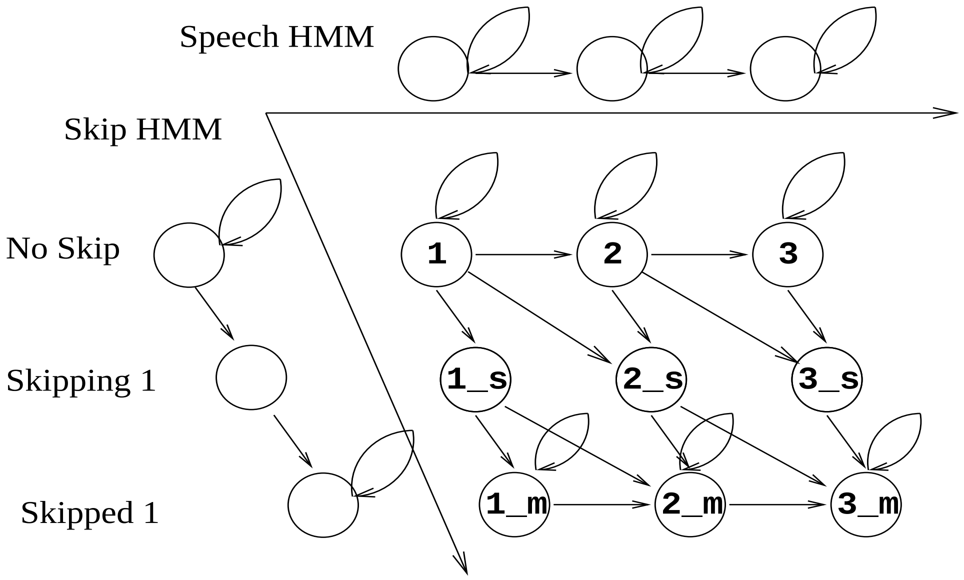

Figure 4.3: The decomposition of an FSHMM into two Markov models.

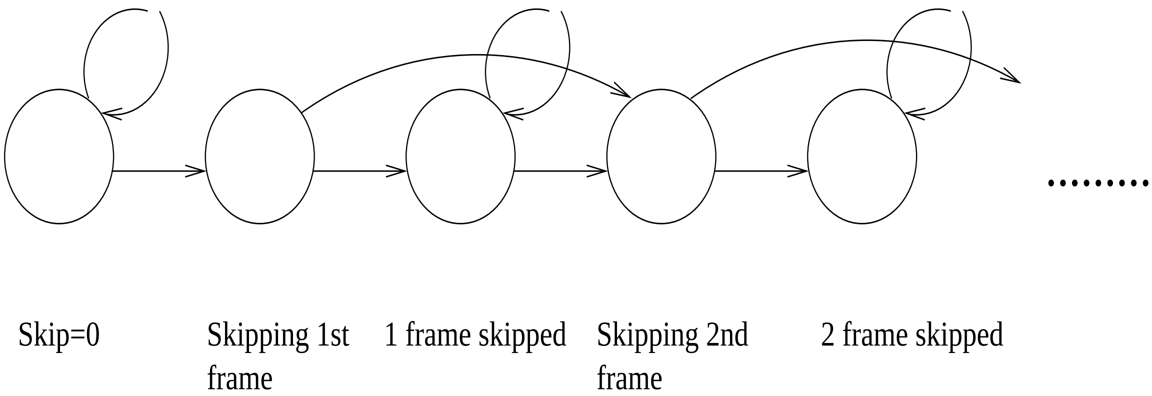

Figure 4.4: The skip-Markov-model with infinite number of skips.

If the number of skips is fixed, the skip Markov model can be written as a finite length chain of state. If the number of skips is infinite, it will be an infinite chain. Similar to reducing the infinite chain of an infinite impulse response (IIR) filter (or all zero form) to a feedback structure (or pole structure), one can redraw the skip Markov model as Figure 4.5.

The above discussion suggests that FSVA of skipping infinite number of frames can be done by using a HMM like Figure 4.6, where all the s states are skip states.

In practice, if we set the observation probability of a skip state in an FSHMM

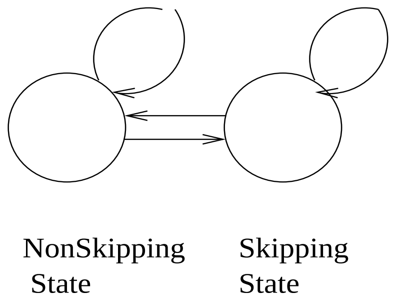

Figure 4.5: Another representation of the skip-Markov-model with infinite number of skips.

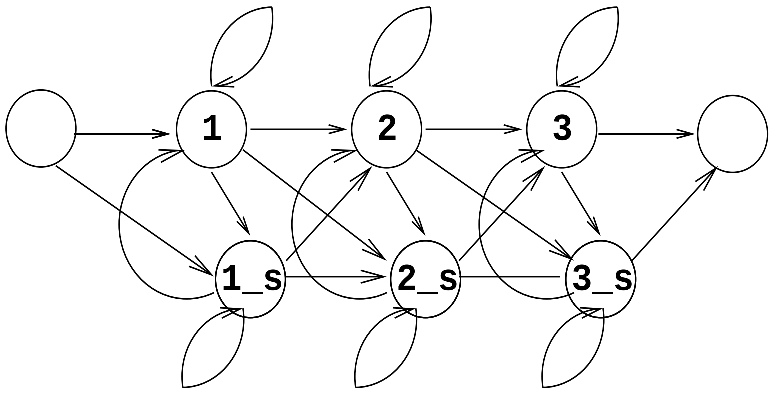

Figure 4.6: The simplified frame-skipping hidden Markov model.

to be 1 using the HMM shown in Figure 4.6, the FSHMM basically ignores all frames in an utterance. Hence, this setting is not useful in practice. However, if we allow some floor values of the observation probability of corrupted frames, this model will selectively skip the frames and make this model can be useful in practice. This involves the problem of tuning of these floor values. We will revisit this problem in Chapter 6.

4.5.2 The Frame-based Extended Union Model Search

HMM-based speech recognition is typically viewed as finding the most likely state sequence $\hat{Q}_1^T$given the observation O_1^T as we described in Section 2.1:

$$\hat{Q}_1^T = \arg\max_{Q_1^T} p(Q_1^T | O_1^T) \tag{4.13}$$

By using the Viterbi algorithm, the above maximization can be done efficiently by computing the partial path score $\delta_t(i)$ that explains the observation O_1^t and $q_t = j$. A recursive relation can be derived as,

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i)a_{ij}\right] b_j(o_t) \quad 1 \leq t < T,$$

where $b_j(o_t)$ is the observation probability for observing o_t at state j.

Equation 4.13 can be written as the product of the observation likelihoods and transition probabilities,

$$\hat{Q}_1^T = \arg\max_{Q_1^T} \underbrace{P(\bigwedge_{i=1}^{T} o_i | \; Q_1^T)}_{\text{obs. likelihoods for } Q_1^T} \times \underbrace{\prod_t p(q_t|q_{t-1})}_{\text{transition prob}} \tag{4.14}$$

We can interpret the observation likelihood as the joint probability of observations. From discussion in Section 3.3.3, Equation 4.14 can be more robust to M-frame corruption to short-time noise if we replace the joint probability with the probability of EUM of order M, i.e.

$$\hat{Q}_1^T = \arg\max_{Q_1^T}(P(\mathring{O}_{\vee(T)}^M | \; Q_1^T) \times \prod_t p(q_t|q_{t-1})) \tag{4.15}$$

This effectively means a search for the state sequence, which maximizes EUM with order M. Notice that we only assume the observation likelihood is corrupted but retain the contribution of state transitions. This preserves the duration information. We call this search as **frame-based EUM search**.

The frame-based EUM search requires solutions for two computational intensive problems. First, as we know that typical sentences have number of frames in an order of 10^2. Thus, from discussion in Section 3.3.3, we know that computation of EUM given the state sequence is very intensive if we do it in directly. Second, frame-based EUM is applied to every state sequence in Equation 4.15.

We can see here the formulation of using the EUM in speech recognition has many similarities as that using the robust likelihood.

4.5.3 The Frame-based EUM Viterbi algorithm (FEVA)

As we discussed in the last section, the frame-based EUM search is another way to deal with partial and temporal corruption (PTC). As we can find the efficient algorithm for search of the maximally robust state sequence, it is natural for us to ask whether similar search technique can be applied for finding the state sequence with maximum probability of EUM.

In this section, we describe how similar search technique in FSVA can be used to implement an efficient algorithm of the frame-based EUM search. We called it the frame-based EUM Viterbi algorithm (FEVA).

As we prove in Appendix B, the EUM probabilities of $t-1$ and t frames are actually related as follows,

$$\begin{aligned} P(\mathring{O}^M_{\vee(t)}) &= P(\mathring{O}^M_{\vee(t-1)})P(o_t) + P(\mathring{A}^{M-1}_{\vee(t-1)}), \\ &- P(\mathring{O}^M_{\vee(t-1)})P(\mathring{O}^{M-1}_{\vee(t-1)})P(o_t), \quad (4.16) \\ &\approx P(\mathring{O}^M_{\vee(t-1)})P(o_N) + P(\mathring{O}^{M-1}_{\vee(t-1)}). \quad (4.17) \end{aligned}$$

where $\mathring{O}^M_{\vee(t)}$ express the event of t-corruptions occurred in M frames. We called this **extended union model formula** or **EUM formula**.

The EUM formula implies that the frame-based EUM search can be efficiently carried out. We define the partial path score that explains the first t observations with order M (M frames are corrupted) at state j as $\delta_j(t, M)$. To find the best state sequence for order M, at each state, $M+1$ different partial path scores are stored for union model of orders $0, 1, \ldots, M$.

Using the EUM formula, we can compute $\delta_j(t, m)$ for all $0 \leq m \leq M$,

$$\begin{aligned} \delta_j(t, m) &= \max_i a_{ij}[\delta_i(t-1, m)b_j(o_t) + \delta_i(t-1, m-1) \\ &- \delta_i(t-1, m)\delta_i(t-1, m-1))], \quad (4.18) \\ &\approx \max_i a_{ij}[\delta_i(t-1, m)b_j(o_t) + \delta_i(t-1, m-1)]. \quad (4.19) \end{aligned}$$

The approximation in Equation 4.19 omits contribution of the product term. We call the Equation 4.18 as frame-based EUM Viterbi algorithm (FEVA).

Similar to the FSVA, the FEVA can be implemented as a N-best like manners. Figure 4.7 shows the implementation of the approximated form of FEVA. At each box, one stores the extended state information which contains the partial path scores for union models with different orders. The j_m-th box represents the maximum union probability with order m at state j. The original Viterbi

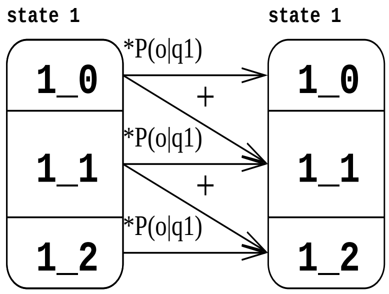

Figure 4.7: The update diagram for an approximated FEVA in a single-state HMM.

path are stored in each $*\_0$ box. In an update, we collect the scores from all possible incoming arcs calculated from Equation 4.18 and take the maximum among them.

Similar to the Viterbi algorithm, a backtrack pointer is used to record the state with maximum partial scores for each state. This allows us to find the best state sequence by backtracking. Unlike the FSVA, we cannot determine the best previous order index for every order because Equation 4.18 sum all the contributions of probabilities of EUM with two different orders, m and $m-1$. Hence, the FEVA can only search for a single state sequence even the order of the search is larger than 0.

It is possible that Equation 4.18 requires computation of union probabilities with either $M < 0$ or $T < 0$ or $M > T$. In all these cases, we can take $P(\mathring{O}^M_{\vee(T)}) = 0$ as we discussed in Section 3.3.3.

The computation and memory size of the FEVA is M times larger than the standard Viterbi algorithm if we find the probability of EUM of order M. With the use of caching, the computation could be similar to that of the Viterbi algorithm. Of course, similar to FSVA, the computation of Equation 4.18 can further be reduced if we omit the contribution of the cross-product term.

In this section, we find that the FEVA can be formulated and implemented in a similar manner to the FSVA. It is natural to ask what are the differences between the two searches. We will describe this in next section.

4.5.4 Relationship between the FSVA and the FEVA

If we compare the formulations of the FSVA and the FEVA, we will see that they are very similar.

$$\begin{aligned}
&\text{Frame-based EUM search :}\\
\hat{Q}_1^T &= \arg\max_{Q_1^T}(\phi_K(\, O_1^t|\; Q_1^t) \times \prod_t p(q_t|q_{t-1})). &(4.20)\\
&\text{Robust state sequence search :}\\
\hat{Q}_1^T &= \arg\max_{Q_1^T}(P(\mathring{O}^M_{\vee(T)}|\; Q_1^T) \times \prod_t p(q_t|q_{t-1})). &(4.21)
\end{aligned}$$

They basically search for state sequences with two different criteria. Interestingly, we observe that the two equations are approximately the same. This can be understood by the definition of the EUM search,

$$\begin{aligned}
\hat{Q}_1^T &= \arg\max_{Q_1^T}(P(\mathring{O}^M_{\vee(T)}|\; Q_1^T) \times \prod_t p(q_t|q_{t-1}))\\
&= \arg\max_{Q_1^T}(P(\bigvee_{i=0}^{C_M^T}(o_{i_1} \wedge o_{i_2} \wedge \ldots o_{i_{T-M}}|\; Q_1^T) \times \prod_t p(q_t|q_{t-1}))\\
&\approx \arg\max_{Q_1^T}(\sum_{i=0}^{C_M^T} P(o_{i_1})P(o_{i_2})\ldots P(o_{i_{T-M}}) \times \prod_t p(q_t|q_{t-1})) &(4.22)\\
&\approx \arg\max_{Q_1^T}(\max_{i=0}^{C_M^T} P(o_{i_1})P(o_{i_2})\ldots P(o_{i_{T-M}}) \times \prod_t p(q_t|q_{t-1})) &(4.23)
\end{aligned}$$

Equation 4.23 follows from Equation 4.22 because of the fact that the value of likelihood of corrupted frames is typically very low. Thus, those terms involve corrupted observations can be generally omitted.

Hence, the FSVA with M frames skipped is approximately equivalent to the FEVA of order M. This is the reason why in this thesis, all evaluations are done for FSVA only. However, we should notice that we have not evaluated the FEVA in full detail in this thesis, which one is better is actually an open question. We may argue that the FSVA has an algorithmic advantage. As we have mentioned in last section, the FEVA can only generate single solution even if the order of the search is larger than 0. The K-solutions of the FSVA can be generated concurrently. This special property of the FSVA actually allows us to select number of skip easily. This topic will be discussed in Chapter 6.

4.6 Summary

In this chapter, we described an algorithm that can against partial and temporal corruption (PTC). We formulated this idea by omitting the contributions of corruption and developed an algorithm called frame-skipping Viterbi algorithm (FSVA). We describe two possible implementations of the FSVA, one is a direct implementation of the update formula by expanding the state space and one expands the topology of HMM to achieve the same purpose.

We establish relations between the FSVA with the SND. We find that the SND interpretation gives us another perspective to view our problem. We see that how similar search technique of the FSVA can be used to implement an efficient search of state sequence with maximum EUM probability. Finally, we found that the FSVA and the FEVA are approximately equivalent to each other.

Chapter 5

Initial Study: Evaluation Results in Gaussian Replacement Noise

We described the general theory of the frame-skipping Viterbi algorithm (FSVA) in the previous chapter. In this chapter, we report the evaluation of the FSVA under the environment that some frames in an utterance are randomly replaced by Gaussian noise. There are several reasons for choosing this type of corruption to evaluate the FSVA. First of all, replacements rather than addition of Gaussian noises ensure that the corrupted frames are non-speech in nature. Hence, it is highly likely that this corrupted frames will be skipped in the FSVA. For the FSVA, it would be a simpler task to start. This gives us a more convenient environment that allows us to focus on the algorithmic properties of the FSVA. Second, replacement noise simulates the phenomena of frame loss in speech communication in network environment.

We first describe the experimental setup in Section 5.1. This includes a description of the corpus and the training and testing conditions. We evaluate the FSVA in clean utterances without corruption that serves as our baseline result in Section 5.2. Then, we evaluate FSVA in utterances with frames which are randomly replaced by Gaussian noise (Section 5.3). One very natural question to ask is that whether FSVA can really skip those corrupted frames. We give an analysis to answer this question in Section 5.3.3. Based on the experimental results and analysis, we can then describe some limitations of FSVA in Section 5.4. Finally, we summarize this chapter in Section 5.5.

5.1 Experiment Set-up

Throughout this chapter, we use the TI-digits corpus [34] for evaluation. The TI-digits is a speaker-independent connected-digit recognition corpus in which every speaker is required to speak about 77 digits sequences. There are several reasons for using TI-digits.

1. Availability of the corpus

   The TI-digits can be easily obtained from Linguistic Data Consortium (LDC). This makes it possible for other researchers to repeat our experiment.

2. Recognition

   The TI-digits has already been widely used by other researchers to evaluate several noisy speech recognition techniques. For example, J. Ming *et. al.* reported results on TI-digits for his work on the extended union model in [42]

3. Simplicity of the task

   The digit recognition task is relatively small as compare to other tasks such as dictation. It is also possible to obtain acceptable performance with limited computational resources. Other task such as dictation requires tremendous computation to get comparable performance to the reported result in the literature.

Our training makes use of HTK [67]. The FSVA is implemented by our own. Their details are described in Section 5.1.1 and 5.1.2 respectively.

5.1.1 Training Conditions

We represented every digit (0-9 and 0 is represented by both "oh" and "zero") as an 18-state left-to-right HMM and the topologies of silence and short pause are the same as those described in the standard AURORA's evaluation [22]. Our choice of models' topologies is based on the fact that a digit can be quite lengthy in the TI-digits corpus. Thus using more state is a better choice than the standard 3-state topology as in [52]. We used HTK in our training process and had followed the training procedure described in [22] to estimate the HMMs' parameters. The training data set consisted of a total of 8668 adult sentences.

Speech was parametized using 12 mel-frequency cepstral coefficients, the energy, their delta and delta-delta coefficients. There were a total of 39 parameters in a feature vector. The detail description is given in the Appendix D.

5.1.2 Evaluation Conditions

We had implemented the frame-skipping Viterbi algorithm (FSVA) for evaluation, which takes an arbitrary network in HTK-ARPA format [67] and HTK acoustic models.

Throughout our study, we used a simple left-to-right network for isolated-digit recognition (IDR) as shown in Figure 5.1. The evaluation data set consists of a total of 8668 adult sentences that do not overlap with the training set. All results shown are in word accuracies. The definition of word accuracy can be found in Section 2.2. We had omitted silence and short-pause in our evaluation as a common practice in evaluation of speech recognition.

5.2 Evaluation in Clean Condition

This section describes the evaluation result of FSVA in clean condition. As the FSVA uses fewer frames in making decision than the standard Viterbi algorithm, results obtained from FSVA may be potentially inferior to that of a normal Viterbi algorithm.

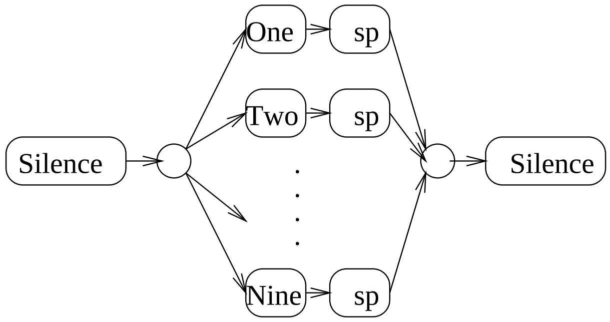

Figure 5.1: The word network for isolated digit recognition.

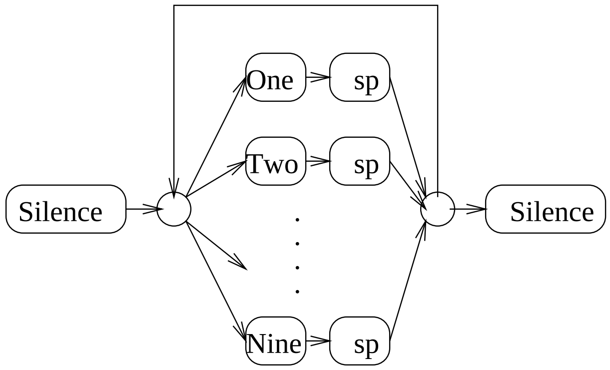

Figure 5.2: The word network for connected digit recognition.

We first evaluated the recognition accuracy obtained by using the standard Viterbi algorithm. In IDR and CDR, the accuracy is 99.72% and 98.95% respectively.

The word accuracies of both IDR and CDR are plotted against the number of skips in Figure 5.3. We then evaluated the performance of the FSVA as a function of the number of frames skipped. Note that zero skip is the same as using the standard Viterbi algorithm. In this experiment, we found that the FSVA's performance was similar to that of the standard Viterbi algorithm when the number of skips is smaller than 5. However, the accuracy decreases when the number of frames skipped increases. This experiment shows that if we can pick up the correct number of skips, the FSVA can have comparable performance as the standard Viterbi algorithm in clean speech. However, if the number of skips is too much, it can degrade performance.

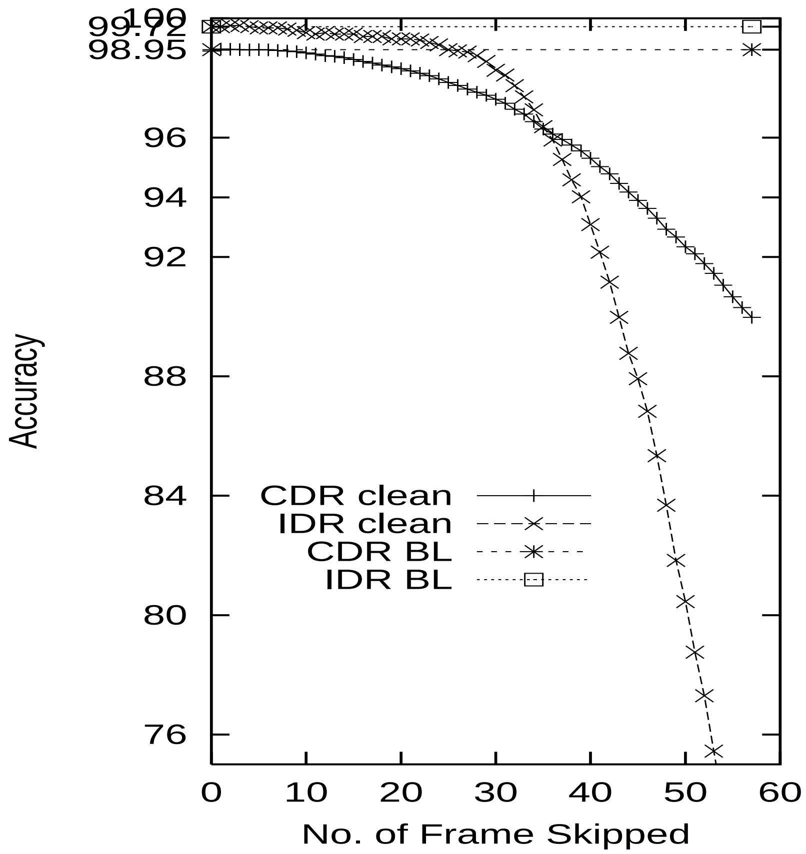

Figure 5.3: The evaluation of the FSVA in clean condition. Note that zero skip is the same as using the Viterbi algorithm.

5.3 Noisy Utterance Evaluation: Frames Replaced by White Noise

We evaluated the FSVA's performance in utterance corrupted by white noise in this experiment. The detail corruption process, i.e., how we add the Gaussian noise to the speech signal is described in Section 5.3.1. The evaluation and analysis section are presented in Section 5.3.2 and Section 5.3.3 respectively.

5.3.1 Description of Noise Addition

We simulated the effect of impulse noise with time-invariant spectral characteristic in this experiment by randomly replacing frames in test utterances by white Gaussian noise. The procedure is as follows,

1. Divide the digital speech signals into non-overlapping blocks[1]

   Each block contains 500 samples and blocks are non-overlapped.

2. For each block,

   a, generates a uniformly distributed random number in the range of [0,1],

   b, if the random number is smaller than 0.1,

   replace the block by a Gaussian noise block with constant magnitude.

The Gaussian noise block is generated by the central limit theorem method described in [2].

5.3.2 Evaluation Result

Using the procedure described in Section 5.3.1, we corrupted in average of 10% of speech frames with Gaussian noise for the whole corpus. We first evaluate the performance of decoding using the standard Viterbi algorithm. The baseline result for IDR and CDR tasks are 85.38% and 78.66% respectively. Then, the FSVA was evaluated with different number of skips. The results are plotted in Figure 5.4.

From Figure 5.4, we can select the optimal number of skip for the two different tasks and the optimal results are summarized in Table 5.1.

The FSVA can significantly improve the recognition accuracy in noisy conditions. The maximum relative gain is 88% and 70% for IDR and CDR tasks respectively. As demonstrated in Figure 5.4, the selection of the number of skips is an important issue. Skipping too few frames probably cannot eliminate the influence of corrupted frames while skipping too many frames removes too much information that deteriorates the performance. Worst of all, the optimal number of skips is task-dependent. In the clean speech task we presented in last section, we find that the optimal number of skips is less that 5. However, in this task, the

[1]Throughout this section, we will use the term *block* to describe *frame in noise addition*. Later, when we describe the recognition setting, we will use the term *frame* to describe the *frame in recognition*. This choice of terms avoid reader's confusion.

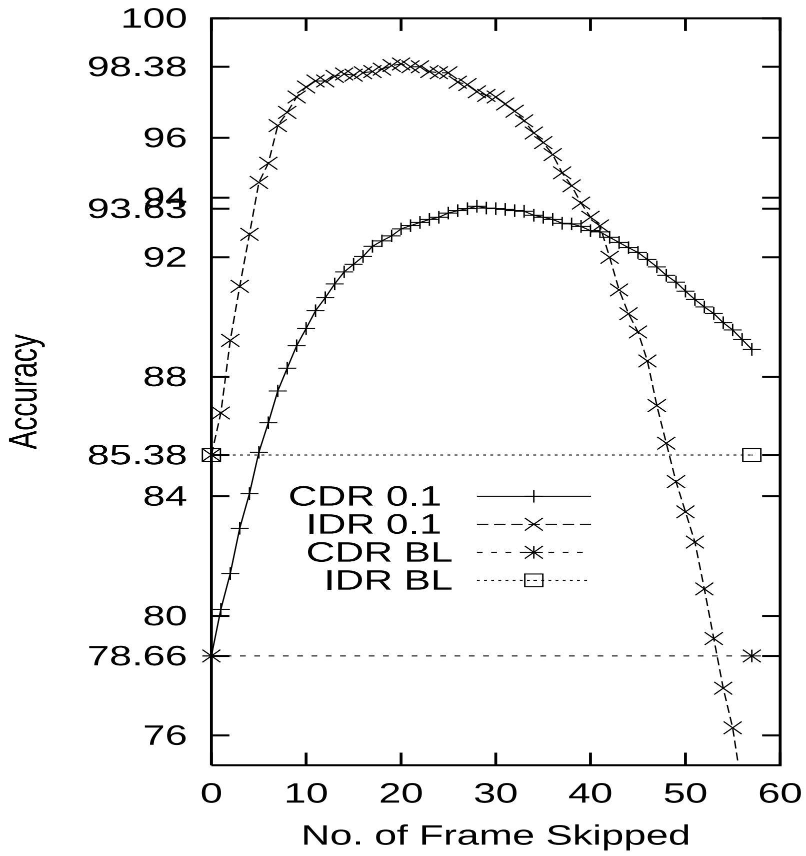

Figure 5.4: The Evaluation of the FSVA in testing utterances replaced by 10% of white noise.

optimal number of skips is found to be 20 and 28 for IDR and CDR respectively. The number of skips is larger in CDR because in the corpus, a connected digit sequence with more than 1 digit is also longer in terms of time duration and with a fixed number percentage of corruption would require skipping more frames.

5.3.3 What Kind of Frames Does the FSVA Skip?

We have shown that the FSVA performed significantly better than the standard Viterbi algorithm in speech corrupted by short-time Gaussian noise. What kind of frames does the FSVA skip? Does it skip those corrupted frames? To answer the above questions, we recorded those frames that are corrupted by the replacement noise and denoted the set as A. For frames that are skipped in FSVA, we denoted them as S. Thus, for a given number of skips in an evaluation, we can

Condition	Acc	No. Fr.
cont. & noisy	93.71	28
cont. & clean	98.97	2
iso. & noisy	98.47	20
iso. & clean	99.76	2

Table 5.1: The best possible performance of the FSVA, which can be achieved in various conditions by changing the number of skips. The last column shows the skip that can achieve optimal value.

always estimate the probability of skipping a corrupted frame,

$$P(S|A) = \frac{\text{no of frames skipped and it is corrupted}}{\text{no of frames which is corrupted}}.$$

and the probability of skipping a clean frame or uncorrupted frames,

$$P(S|\tilde{A}) = \frac{\text{no of frames skipped and it is not corrupted}}{\text{no of frames which is not corrupted}}.$$

To evaluate whether our skipping algorithm is targeting the corrupted frames, we compare the value of $P(S|A)$ and $P(S|\tilde{A})$,

$$\text{Likelihood Ratio} = \frac{P(S|A)}{P(S|\tilde{A})}.$$

The larger the likelihood ratio indicates that the skipping algorithm would tend to skip the corrupted frame more often than those uncorrupted frames. If the likelihood ratio is equal to 1 would mean that the FSVA does not discriminate between the corrupted frames and the uncorrupted frames.

The measurement of the sizes of the above sets are complicated by different settings of the noise addition and the recognition. To avoid confusion, we called a *frame in noise addition* as a *block* and continue to represent a *frame in recognition* by a *frame*. In our experiment, the noise addition are done by randomly replacing non-overlapping blocks by Gaussian noise. However, the recognition decodes on overlapping frames. Hence, one corrupted *block in the noise addition* actually corrupts multiple *frames in the recognition.*

How many frames are corrupted by one block of noise ? We can answer this question by comparing the settings in detail. In noise addition, the utterance is divided into non-overlapping blocks and each of them consists of 500 speech samples (25ms). In recognition, one frame also consists of 500 speech samples (25ms) but they overlap. Two consecutive frames have 300 samples (15ms) of overlapping when the analysis window is actually shifted by 200 samples (10ms) every time (refer to Appendix D). This is illustrated in Figure 5.5

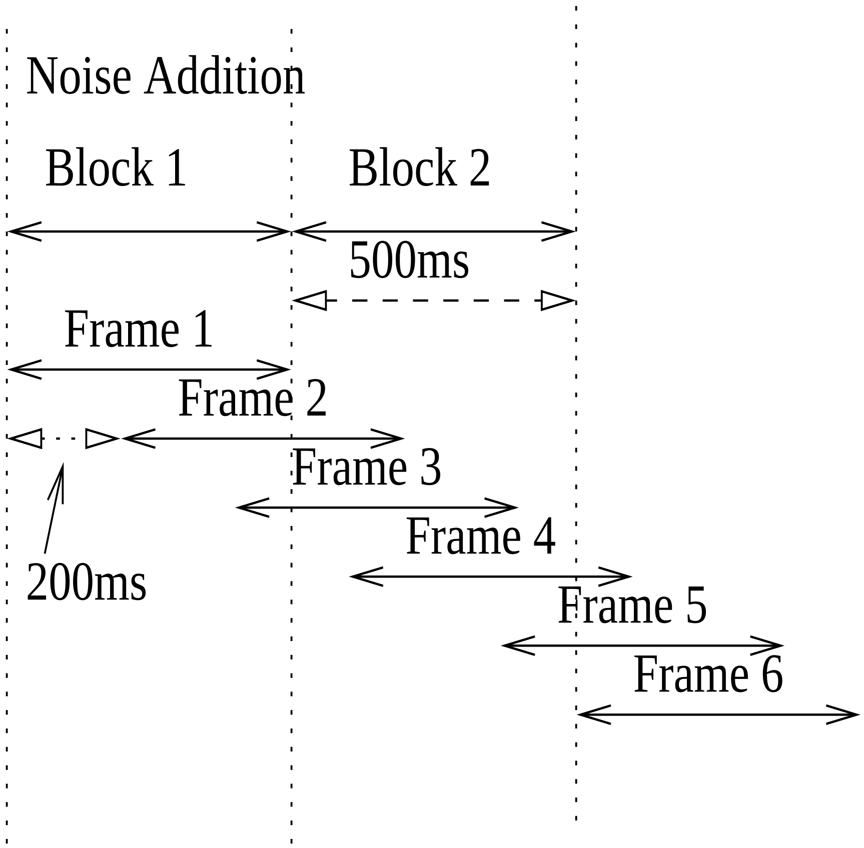

Figure 5.5: The relation between blocks and frames.

We observe several facts in Figure 5.5,

- Two blocks can be converted to 5 frames in our experiment. For example, if an utterance contains 100 blocks measured in the noise addition, there will be 250 frames in the recognition. This rule of thumb allows us easy to convert the total number of blocks to total number of frames in an utterance.

- If a block is corrupted, depends on the block is, it would corrupt different number of frames. The general rules are as follow.

  If the block number is odd, it will corrupt 9 frames with 1 frame totally corrupted, 2 frames 60% corrupted, 2 frame 20% corrupted and 4 more frames with corrupted delta and delta delta coefficients.

  If the block number is even, it will corrupt 8 frames with 2 frames 80% corrupted and 2 frames 40% corrupted and 4 more frames with corrupted delta and delta delta coefficients.

  If we neglect the effect of the corrupted delta and delta delta coefficients, in average 1 corrupted blocks will cause 4.5 frames to be corrupted.

  Hence, in our experiment, because 10% of blocks are corrupted, there will be about 18% of frames are corrupted.

Based on the above observation, we can present our measurement here. We recorded that there are totally 606846 blocks in the corpus. Hence, there are totally $606846 * 5/2 = 1517115$ frames. As 10% of blocks are corrupted, that means there are totally $606486 * 10\% = 60684$ blocks are corrupted. Hence, in average $60684*9/2 = 276400$ corrupted frame and there are 1240815 uncorrupted frames.

Hence, the actual formula we are using in computing the log likelihood ratio would be,

$$LR = \frac{\frac{\text{no. of skipped and corrupted frame}}{276300}}{\frac{\text{no. of skipped and uncorrupted. frames}}{1240815}} \tag{5.1}$$

Based on Equation 5.1, we plot the likelihood ratio in Figure 5.6. In the graph, we see that the likelihood ratio is always larger than 1. Hence, we can believe that FSVA really skip those corrupted frames.

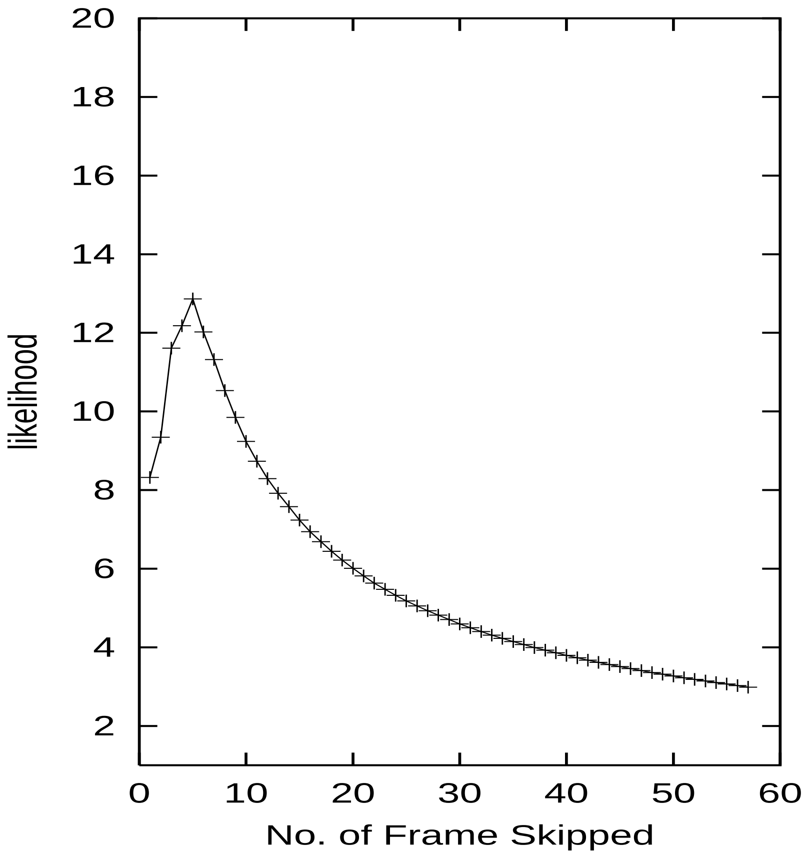

Figure 5.6: The log likelihood ratio against the number of skips in testing utterances replaced by 10% of white noise.

5.4 Limitations of Frame-Skipping Viterbi Algorithm

From the discussion in the above sections, we can see that the FSVA performs satisfactorily in recognizing speech corrupted by random Gaussian noise replacement. However, there are several limitations and was explicitly shown in our experiments.

1. **Difficulty in determining the number of skips**

   It is very difficult to determine the value of K, the number of frame to skip, in practice. This will be a very important issue because skipping too many frames will cause degradation of the quality of the decision. However, skipping too little weaken the effect of the FSVA. This situation is shown in the experiment in Section 5.3 where the optimal skip is dependent on the environment.

2. **Memory**

   We see that in our implementation, the amount of memory used is K times larger than a normal Viterbi algorithm.

We try to solve the above problems by developing two variants of FSVA in the next chapter.

5.5 Summary

In this section, we have shown that the FSVA can be used in speech recognition against noise that randomly replaces speech frames. It is shown to have comparable performance in clean condition as the standard Viterbi algorithm. It outperforms the Viterbi algorithm in noisy condition provided that the number of skips can be determined. The analysis section also showed that the FSVA is more likely to skip those frames that are corrupted.

Chapter 6

Improvements of the FSVA

We described the theory and the implementation of the frame-skipping Viterbi algorithm (FSVA) in Chapter 4 and evaluated its performance in Gaussian replacement noise in Chapter 5. In Section 5.4, we found two limitations in the FSVA. First of all, it is difficult to determine the optimal number of skips for a sentence. If we use a strategy such as skipping a fixed number of frames, the performance is easily affected by factors such as the length of a sentence and the rate of corruption. Second, we observe that the FSVA is a memory-intensive algorithm that uses $k+1$-times memory of a standard Viterbi algorithm. These two problems limit the use of the FSVA in real-life situation

In this chapter, we seek for a mechanism that can automatically determine the number of skips, K, in the FSVA. We call any algorithm that can achieve this purpose as **automatic skip determination (ASD)** algorithm. We describe a method called the log-likelihood ratio thresholding (LLRT) as one solution of the ASD problem. The motivation, the theory and the implementation of LLRT are described in Section 6.1 and evaluated it in Section 6.2.

The success of the LLRT sheds light on how we can reduce the amount of the memory used in the FSVA. We called the memory-efficient algorithm as the simplified frame-skipping Viterbi algorithm (SFSVA). We only provide a preliminary study and evaluation in this thesis. We describe it in the conclusion(Section 6.3) of this section.

6.1 Strategy of the Automatic Skip Determination: Log Likelihood Ratio Thresholding (LLRT)

In the last chapter, when we evaluate the FSVA in different conditions, different optimal number of skips, K, is required. It is intuitive that if we skip different number of frames for every sentence, the performance will improve. This is because the length of utterances and the rate of corruption in every sentence is different.

If we have prior knowledge of the noise environment, solving the ASD problem can follow some simple strategies such as skipping a fixed number of frames in an utterance or skipping a fixed proportion of frames in an utterance. However, as the noise condition is usually unknown, it is difficult to determine the parameters of the above simple strategies.

Ideally, what we seek for is a mechanism that does not require any knowledge of the noise environment. In this section, we describe one such method called the log likelihood ratio thresholding (LLRT) that can satisfy this purpose. Analogous to stopping of tree growing in the use of decision-tree, we found that a threshold can be used in stopping the skipping process of FSVA. Interestingly, this threshold can be thought as the minimum likelihood of the clean frame. This simple method yields surprising good performance in different environments. More importantly, the performance is insensitive to the parameter.

The use of the LLRT in determining the number of skips is made possible by the fact that the FSVA can generate multiple solutions when skipping K frames. We first describe this fact and the relation between solutions with different skips, K, in Section 6.1.1. This serves as the theory of our solutions. The method is described in Sec 6.1.2.

6.1.1 Multiple Solutions Generated by the FSVA

We have mentioned in Section 4.3 that a K-skip FSVA can generate $K+1$ solutions. This comes from the recursive formula in Equation 4.7 or can be visualized by the K-skip frame-skipping hidden Markov model (FSHMM) as in Figure 6.1. One can see that a K-skip FSHMM has $K+1$ ending nodes. Hence, $K+1$ paths can be obtained by back-tracking from each node. Each node represents the maximally robust state sequence of different skips. They have likelihoods with values $(\phi_0(\ O_1^T|Q) \ldots \phi_K(\ O_1^T|Q))$ as we have defined in Section 4.3.

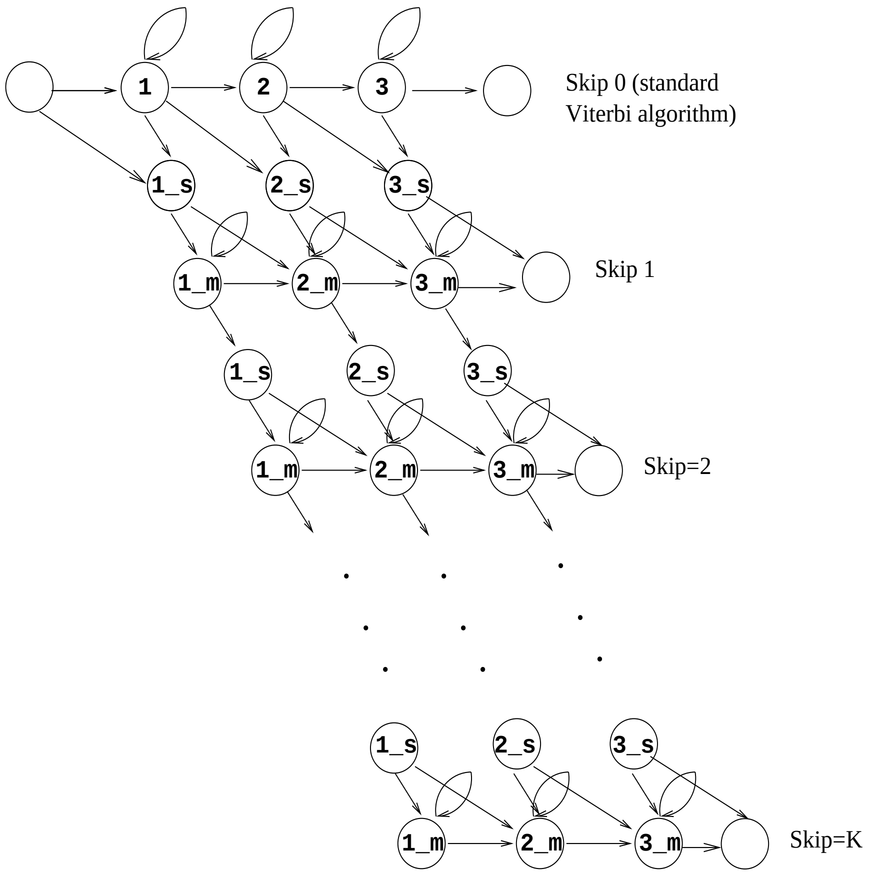

Figure 6.1: The topology of an FSHMM with K skips.

It is interesting to understand the relationship between the different maximally robust state sequences with different skips. From Equation 4.3 described in Section 4.

$$\phi_K(\ O_1^T|Q) \quad = \quad \phi_{K+1}(\ O_1^T|Q) + \log p(o_{t_{K+1}}|q_{t_{K+1}}),$$

$$\leq \phi_{K+1}(O_1^T|Q). \tag{6.1}$$

Based on this result and from the definition of robust state sequence with K skips, we have,

$$\begin{aligned}
\phi_{K+1}(O_1^T|\hat{Q}_1^T(K+1)) &\geq \phi_{K+1}(O_1^T|Q), \quad \forall Q \\
&\geq \phi_{K+1}(O_1^T|\hat{Q}_1^T(K)), \qquad (6.2) \\
&\geq \phi_{K}(O_1^T|\hat{Q}_1^T(K)). \qquad (6.3)
\end{aligned}$$

Here $\hat{Q}_1^T(K)$ is the K-skip maximally robust state sequence. Hence, the robust likelihood score is increasing monotonically.

Analogous to the iterative decision tree growing [68][69] in which the training likelihood always increases as the tree grows, the frame-skipping can be thought as an process which interatively increases the criterion function of the process. In decision tree growing, one stopping criterion is to threshold on the gain in likelihood. This analogy inspires us to devise a method to determining the size of the appropriate number of skips. We describe this method in next section (Section 6.1.2).

6.1.2 Log Likelihood Ratio Thresholding (LLRT)

As the robust log likelihood increases as K increases, an intuitive solution is to compute the relative change in the robust likelihood. Then, we compare this change with a threshold $\log c$.

$$\phi_K(O_1^T|\hat{Q}_1^T(K)) - \phi_{K+1}(O_1^T|\hat{Q}_1^T(K+1)) > \log c. \tag{6.4}$$

Since $\phi_K(O_1^T|\hat{Q}_1^T(K))$ is the K-skip robust likelihood, we denote the left-hand-side of Equation 6.4 as $LLR(K, K+1)$ and Equation 6.4 as the log likelihood ratio (LLR). Figure 6.2 shows a typical plot of $LLR(K, K+1)$ against K. We observe that there is a general decreasing trend of $LLR(K, K+1)$. The value of c has the intuitive meaning of the minimum likelihood of a clean frame and this is a function of the model and training conditions.

In the actual implementation, we can first generate solutions for K skips and select the best solution using the criterion in Equation 6.4.

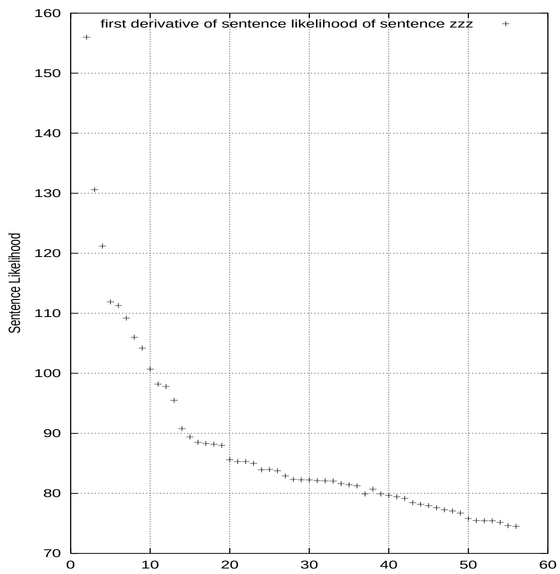

Figure 6.2: The log-likelihood ratio of utterance “zzz” against the number of skips.

6.2 Performance of Log Likelihood Ratio Test

In last section (Section 6.1), we described the theory of LLRT. This section provides the evaluation of this method in the same task in Chapter 5. The main target in this section is to evaluate the performance of LLRT in a particular noise condition. Furthermore, we would like to see LLRT changes in different environments and shows the LLRT threshold is invariant across different noise conditions. This section illustrates the superiority of LLRT by comparing it with other simple strategies including skipping a fixed number of frames (FIX) and skipping a fixed percentage of frames (PER). We make use of the four tests described in last chapter (Chapter 5), including isolated digit recognition (IDR) and continuous word recognition (CDR) in clean utterances and utterances corrupted by Gaussian replacement noise.

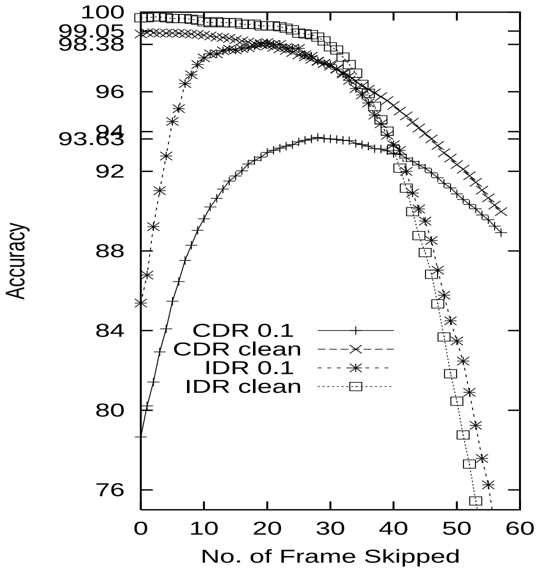

Figure 6.3: The performance of the FSVA against the number of skips using the FIX strategy for automatic skip determination.

6.2.1 The Evaluation of Strategies FIX and PER in Gaussian Replacement Noise

We have evaluated the strategy FIX in last Chapter (Chapter 5). Hence, we mainly review its result and present the results of strategy PER here.

The strategy FIX has only one parameter, which is the number of skips for every utterance. As a summary, we tabulated the evaluation results for all conditions (clean IDR, noisy IDR, clean CDR, noisy CDR) in Figure 6.3 and the optimal results obtained in Table 6.1. From the results, we see that the parameter in this strategy, number of skips, is very sensitive to changes in the rate of corruption. One can see that in clean and noisy conditions, different numbers of skips have to be used. Also, the performance of strategy FIX is also quite sensitive to number of skips.

Next, we describe our evaluation of the strategy that skips a fixed percentage of frames in every sentence (strategy PER). The parameter in this strategy is the percentage of frame skips. The results are summarized in Figure 6.4 and the optimal results obtained in Table 6.2. One can see that it has a similar

Condition	Acc	No. Fr.
cont. & noisy	93.71	28
cont. & clean	98.97	2
iso. & noisy	98.47	20
iso. & clean	99.76	2

Table 6.1: The best possible performance of the FSVA using the strategy FIX that can be achieved in various conditions by changing the number of skips.

problem as the strategy FIX. As one may expect, the optimal percentages used in clean and noisy conditions are different. Hence, the parameter is still sensitive to different environments. One may find that the sensitivity of the strategy PER to the parameter is relatively smaller than the strategy FIX. We may explain this by the fact that the strategy FIX may skip too many frames in short sentences when the number of frame skips is large. However, this is related to the fact that a fixed percentage of corruption is added.

Condition	Acc.	Percentage
cont. & noisy	94.55	14
cont. & clean	98.97	1
iso. & noisy	98.18	15
iso. & clean	99.76	3

Table 6.2: The best possible performance of the FSVA with the strategy PER as automatic skip determination in various conditions, the last column shows the percentage of skips required to get the optimal result.

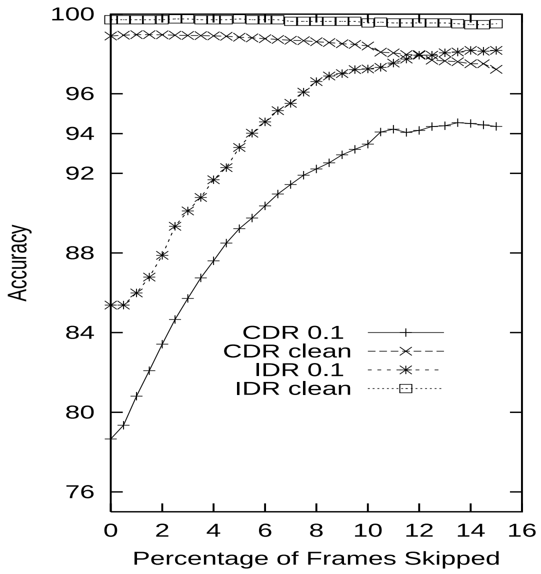

Figure 6.4: The performance of the FSVA against the number of skips using the PER strategy for automatic skip determination.

6.2.2 Evaluation of the LLRT in Gaussian Replacement Noise

Finally, we present the evaluation result of strategy LLRT here. The parameter of this technique is c. As we have explained in Section 6.1.2, the value of c has an intuitive meaning as the minimum likelihood of a clean frame and this can be a function of the model. We showed that the performance against c in Figure 6.5 and the optimal result in Table 6.3. One can see that LLRT has very nice properties in clean conditions. I.e., the performance is unchanged when $\log c > 90$. This means that the sensitivity of the parameter in clean condition is small. Also, one can use the same value of c to achieve undegraded performance in clean conditions and outperform the standard Viterbi algorithm. In Table 6.4, we showed that using $\log c = 90$, the FSVA with LLRT outperforms the standard Viterbi algorithm in all conditions.

The above results show that one might first tune the parameter, c in a single condition and use the same parameter across all conditions.

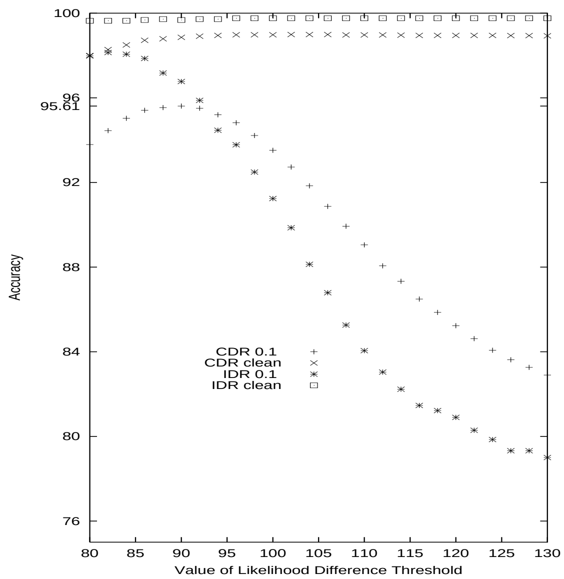

Figure 6.5: The performance of the FSVA against the number of skips using the PER strategy for automatic skip determination.

6.3 Discussion and Conclusions

In this chapter, we described the use of the log likelihood ratio thresholding (LLRT) to automatically determine the optimal number of skips. We showed that the combination of the FSVA with the LLRT outperforms the standard Viterbi algorithm; Its performance is insensitive to changes of the environment and a wide range of values of c can achieve this property. The reason why this method works is that one can use a value c to determine whether a frame is clean or not. Since c has an intuitive meaning as the minimum likelihood of a clean frame, the LLRT is shown to be a good automatic skip determination (ASD) approach. Another implication is that c can be determined from a single condition and be applied to other conditions.

The success of the FSVA with LLRT shed light also on the memory problem of the FSVA. As we have discussed in Section 4.5.1, it is possible to reduce the infinite cascading of FSHMM to a very simple structure. However, we also noticed that without a suitable choice of floor probability of skipped states, the simplified structure virtually skips all the frames in an utterance. The discussion in the FSVA with LLRT suggests that one possible choice of this floor probability

Condition	Acc.	Threshold
cont. & noisy	95.61	90
cont. & clean	98.96	90
iso. & noisy	98.14	82
iso. & clean	99.76	96

Table 6.3: The best possible performance of the FSVA with LLRT as automatic skip determination in various conditions, the last column shows the value $\log c$ that attains the optimal result.

Condition	Acc.
cont. & noisy	95.61
cont. & clean	98.96
iso. & noisy	96.77
iso. & clean	99.68

Table 6.4: The performance of the FSVA that can be achieved in various conditions using the LLRT when $\log c = 90$.

may be taken as the minimum likelihood of clean frames. That is just the value, c, in the LLRT method. Hence, the value determined in the LLRT method can be used as the floor value here. We called the resulting model, as the simplified frame-skipping HMM (SFSHMM).

The SFSHMM can also be implemented using state-space expansion approach, if we floor the values of likelihood, $p(\hat{o_t}|q_t)$, in the following manner,

$$p(\hat{o_t}|q_t) = \begin{cases} b_j(o_t) \text{ if } b_j(o_t) > \log c, \\ \log c \text{ if } b_j(o_t) < \log c, \end{cases} \tag{6.5}$$

We called the Viterbi algorithm that makes use of the above observation as simplified frame-skipping Viterbi algorithm (SFSVA). The use of SFSVA requires no extra memory than a standard Viterbi algorithm and virtually the same computation as it. We describe the pseudo-code at the code-list in next page..

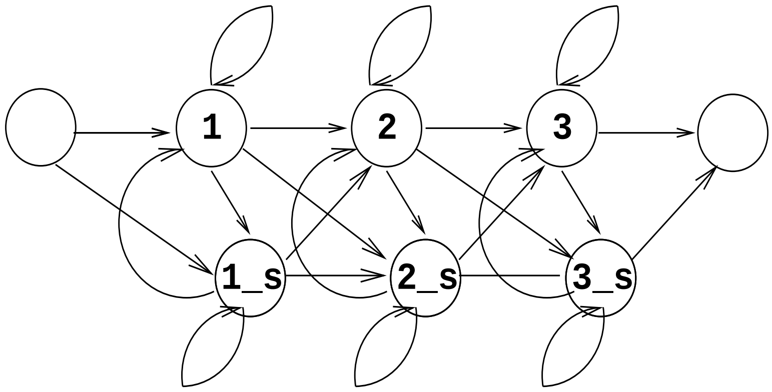

Figure 6.6: The simplified frame-skipping hidden Markov model (SFSHMM).

1. **Initialization** $\gamma_i(1) = \pi_i$ for $1 \leq i \leq N$.

2. **Recursion**

   for t from 1 to $T-1$

   calculate $\alpha_i(t+1)$ base on formula

$$\gamma_i(t+1) = (\max_{j=1}^{N} a_{j,i}\gamma_i(t))\dot{b}_j(o_{t+1})$$
$$\phi_i(t+1) = \arg(\max_{j=1}^{N} a_{j,i}\gamma_i(t)\dot{b}_j(o_{t+1})$$

   where $b_j(o_t)$ is obtained using following formula:

$$\dot{b}_j(o_t) = \begin{cases} b_j(o_t) \text{ if } b_j(o_t) > c, \\ c \text{ if } b_j(o_t) < c. \end{cases}$$

3. **Termination** $P(O|W)$ can be calculated as

$$P(O|W) = \max_{j=1}^{N} \gamma_j(T) \tag{6.6}$$

4. **Back-Tracking**

$$\tilde{s_t} = \phi_{t+1}(\tilde{s_{t+1}}) \quad \text{for } 1 \leq i \leq T$$
$$\tilde{S} = (\tilde{s_1} \ldots \tilde{s_T})$$

We evaluated the SFSVA in Gaussian replacement noise and tabulated the result in Table 6.5. This preliminary study shows that this algorithm outperforms the standard Viterbi algorithm and is comparable to the FSVA with LLRT.

Condition	BL	SFSVA	LLRT
cont. & noisy	78.66	95.38	95.61
iso. & noisy	85.38	98.83	98.83
cont. & clean	98.90	98.73	98.86
iso. & clean	99.72	99.68	99.68

Table 6.5: The performance of the SFSVA that can be achieved in various conditions using $\log c = 90$.

Comparing the FSVA with the FEVA, yet another possible conclusion can be derived from the success of the LLRT method. If we look back to the derivation of LLRT, the fact that the FSVA can generate $K+1$ solutions when searching the K-skip maximally robust likelihood provided the bases for the LLRT such that we can determine the number of skips automatically. In Section 4.5.4, we know that the FEVA cannot generate $K+1$ solutions when and K-th order EUM is assumed. This extra characteristic allows the FSVA to be used to automatically determine the skips by choosing the best from all the available $K+1$ answers. This becomes more difficult for the case of the FEVA because the answers had to be generated one-by-one.

Chapter 7

Evaluation Results of the FSVA in Partial and Temporal Corruption

In Chapters 5 and 6, we evaluated the paradigm of skipping frames in corruptions by Gaussian noise replacement. In this section, we extend the above evaluation to two kinds of partial and temporal corruption. Before we go on, lets review the partial and temporal corruption. We defined what we mean by unknown short-time noise and partial and temporal corruption in Chapter 3 with the following properties,

1. **Random occurrences,**
2. **Random amplitude,**
3. **Random spectral content,**
4. **Random duration.**

   Also, its duration should be shorter than the speech signal

The discussion in Chapters 5 and 6 provides us the theory of the frame skipping Viterbi algorithm (FSVA) (in Section 4.3). The mechanism of automatically determining the number of skips, the log likelihood ratio test (LLRT) is discussed in Section 6.1.

Our previous evaluation of the FSVA in Chapters 5 and 6 focused on corruptions of Gaussian replacement noise. We found that the FSVA with LLRT provides significant improvement of HMM-based speech recognizer in this condition. Gaussian replacement noise does have properties such as random occurrences, amplitude and duration. However, the spectral content is fixed in this kind of corruption and the noise is not additive. Hence, the first part of evaluation extends the replacement noise to additive noises. In these additive noises, their properties are generally controlled by their amplitudes, spectral contents, durations. We would like to show the FSVA with LLRT works regardless of all these variations of conditions and its performance is not sensitive to changes of conditions. This evaluation was done in two tasks. In the first task described in Section 7.1, we randomly select samples of seven noises with different spectral characteristics and add them to the clean speech signal with predefined SNR, duration of noise and rate of corruption. We varied the three factors to understand the properties of performance of the FSVA with LLRT. This allows us to draw conclusion on whether it is useful for this class of noise.

The second task is to extend the result in replacement noise evaluated in Chapter 5 to frame-loss in network speech communication. One can view evaluation in Chapter 5 as a simplified situation of frame-loss in network environment. In reality, codecs in wireless communication such as Global System for Mobile (GSM) and network communication such as Voice Over IP (VoIP) usually specify algorithms to generate frames of “comfort noise” to replace lost frames. This measure can ensure the quality of voice over network. Since the comfort noise is usually generated from the neighborhood speech frames, it becomes a question whether FSVA can still be useful. Hence, we try to simulate the performance of the FSVA with comfort noise generation (CNG) exists in the GSM environment. This evaluation will be described in Section 7.2.

We summarize and conclude the results in this chapter in Section 7.3.

7.1 Evaluation of the FSVA in General Short-Time Additive Noise

In this section, we extend the evaluation of replacement noise in Chapter 5 to additive noise. There are three main goals in this section. Obviously, the first goal is to see whether the FSVA with LLRT work outperforms the standard Viterbi algorithm in this kind of noise. The second goal is to understand how the performance of the FSVA with LLRT varied with the important factors of additive noise. These factors include the amplitude of noise, rate of corruption and the length of corruption. The third goal is to see whether the LLRT's threshold c is sensitive to the changes of the environment. Good performance on this part in effect means FSVA with LLRT can deal with additive short-time noises regardless of the noise environment.

We organize our evaluation as follows, we first describe the experiment set-up in Section 7.1.1. Then, we describe the effect of different factors in Section 7.1.2. The sensitivity of FSVA with LLRT is evaluated in Section 7.1.3.

7.1.1 Experiment Setup

The baseline experiment is the same as that described in Section 5.2. All experiments were performed using the TI-digits corpus. FSVA was evaluated under continuous digit recognition (CDR) with the same HMM topologies used in AURORA baseline experiment [22]. The acoustic features included 12 cepstral coefficients, their energy and the corresponding delta and acceleration coefficients. Only adult sentences of the training and testing sets were used resulting in a total of 8668 digit utterances for training and 8668 for test. Each utterance was a digit string with a maximum length of 7 digits and up to 3 seconds long. We followed the training procedure as suggested in HTK Handbook. The baseline is 98.90% in CDR.

For the evaluation in noisy environment, short-time noise could be characterized by three different attributes.

- **Signal to noise ratio** (S) can be computed as the ratio between the power of speech signal and noise in a sentence. In our research, the signal to noise ratio was computed based on the following formula,

$$SNR = 10 \log_1 0(\frac{\text{Power of the Speech Signal}}{\text{Power of the Noise}}) \tag{7.1}$$

  Here the silence and short-pause were excluded in the calculation in the power of the Speech Signal. As we generated the noise at a frame-by-frame basis, the power of noise is computed by summing the powers of all frames.

- **Rate of corruption** (C) is measured as the percentage of frames corrupted.

- **Length of corruption** (L) is the duration of the noise. We measure in number of frames.

To simulate the situation in which we have no prior knowledge of the noise spectrum, we randomly picked one of the seven types of noise for corruption for each sentence. The seven different noises included 4 telephone ring tones plus 1 whistling noise found in [41], and two message sound clips in the ICQ application. After selecting the noise, we then randomly selected frames (frame length 25ms) from this noise and add it to speech portions of the utterance in a predefined SNR , S, and the rate of corruption , C. This process of noise addition ensured that the noise is unexpected and unknown. One can see the waveforms and spectrograms in Appendix F. The following algorithm shows the details of noise addition.

1. Set the rate of corruptions of short-time noise to value C in the range of [0,1],

2. Set the duration of the short-time noise L. Here L is an integer which represents the number of frames of the noise. One frame contains 500 samples here.

3. Set the noise level to be SNR S,

4. Divide the speech utterances into frame using the specification,

5. For every L frame,

   Randomly select a noise from the noise set. generate a random number in the range of [0,1]

   If the random number is larger than C,

   added the noise to the speech frame with SNR S.

7.1.2 Experimental Results

We evaluated the FSVA with LLRT in additive noise with three different noise conditions and the results are summarized in Tables 7.1, 7.2 and 7.3. The performances were obtained using the best LLRT threshold in each of the condition.

As shown in Tables 7.1, lower SNR caused more degradation to ASR's performance but FSVA is more effective because it might be easier to identify and skip the noisy frames. When $SNR > 0\ dB$, although FSVA was not very effective, it did not degrade performance indicating that FSVA can be used at all times.

In Table 7.2 , we varied the rate of corruption between 20% and 40% respectively at $SNR = -10\ dB$ and the length of noise $L = 1$(the unit is the length of one frame, which is 25ms here). As the rate of corruption increased, the ASR's performance degraded and so was the effectiveness of the FSVA because fewer clean frames were available in the decision process. However, at all rate of

SNR	BL	LLRT (optimal)
∞	98.90	98.99 (102)
10	98.62	98.67 (106)
0	97.57	97.99 (102)
-10	84.04	91.89 (94)

Table 7.1: The performance of the FSVA with LLRT in noise with varying SNR, the rate of corruption, $C = 20\%$ and the length of corruption, $L = 1$. Key: BL: No frames are skipped. LLRT (100): number of skips is determined by the LLRT using threshold optimal value of c (The number inside the bracket shows the value of $\log c$). $SNR = \infty\ dB$ illustrates the performance of strategies in clean utterances.

corruption, FSVA together with LLRT consistently improved performance.

C	BL	LLRT (optimal)
20%	84.04	91.89 (94)
30%	69.21	82.96 (94)
40%	56.39	71.15 (94)

Table 7.2: The Performance of the FSVA with LLRT in noise with varying rate of corruption and $SNR = -10\ dB$ and the length of corruption, $L = 1$. Key: C: The rate of corruption. BL: The performance when no frame are skipped. LLRT (optimal) number of skips is determined by LLRT with the optimal value of $\log c$ (The number inside the bracket shows the value of $\log c$).

The results for varying the length of corruption were shown in Table 7.3 from 1 to 8 frames in which we kept $C = 20\%$ and $SNR = -10\ dB$. Shorter length of corruption caused more damage to the utterances if the rate of corruption was kept constant because it affected more phonetic units making recognition harder.

The results in this section show that the FSVA with LLRT generally works in all conditions provided that we can choose the optimal parameter. In next section, we will show whether this is strictly necessary.

length	BL	LLRT (optimal)
1	84.04	91.89 (94)
2	87.33	91.99 (100)
3	90.24	94.07 (94)
4	93.30	96.22 (92)
5	95.17	97.09 (94)
6	95.69	97.31 (92)
7	95.49	97.22 (94)
8	95.38	96.85 (94)

Table 7.3: The performance of the FSVA with LLRT in noise with varying length, $SNR = -10\ dB$ and the rate of corruptionm, $C = 20\%$. The third column describes the evaluation result of LLRT with the optimal value of $\log c$ (The number inside the bracket shows $\log c$).

7.1.3 Sensitivity of the FSVA with LLRT

In experiments of Section 7.1.2, the optimal LLRT thresholds were used. It is very important to see whether this is necessary. The performance in different conditions are tabulated in Tables 7.1, 7.2 and 7.3 by using a single fixed threshold for all conditions. Figure 7.1 shows the average performance over different noisy conditions. The dashed line is the baseline performance. The solid line is the accuracy obtained using various LLRT thresholds across all conditions. The dotted line is the optimal performance using the optimal value of c in all conditions. The plot shows that there is a wide range of values of c that can improve recognition performance over the baseline. Hence, the performance of FSVA with LLRT is fairly insensitive to the parameter c. More importantly, the average performance using any values from the range is close to the optimal

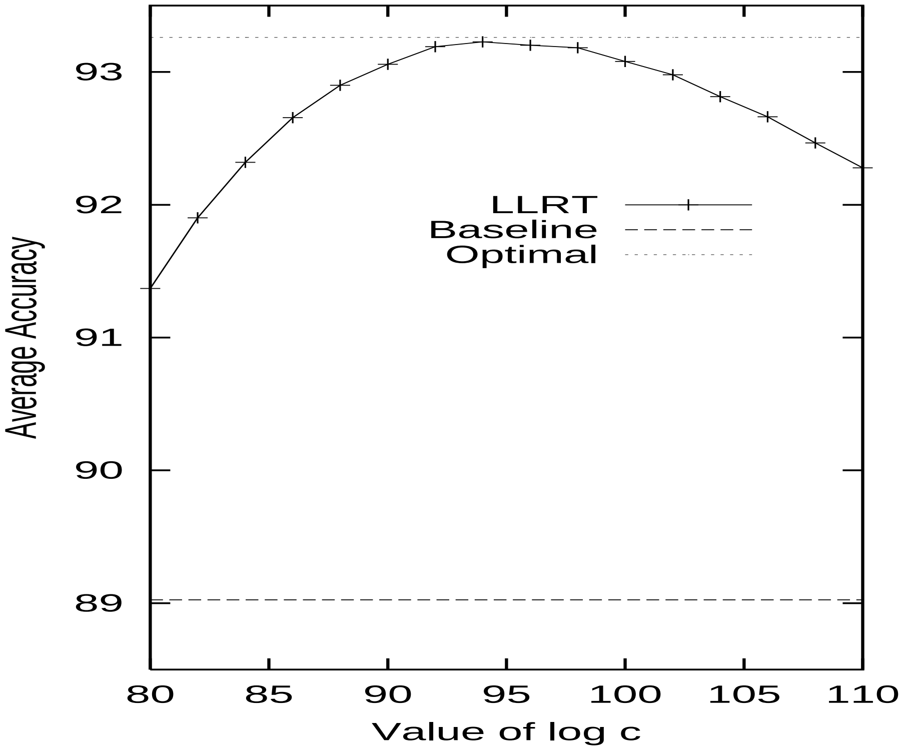

Figure 7.1: The average accuracy of all conditions against the value of $\log c$ in short-time additive noise.

performance obtained by using different LLRT thresholds for different environments and significantly better than the baseline conditions. For example, we could use any threshold from range $(90, 100)$ for all conditions and this will give only slight difference from the optimal performance. This also suggests that the performance of FSVA with LLRT is also insensitive to the changes of the noise environments.

As we have discussed in Section 6.1, the threshold c has the meaning of the minimum likelihood for a clean frame and is dependent on the models. The above results suggest that by tuning the value of c in a single condition, we can obtain a reasonable range of value to use under different noise environments.

7.2 Evaluation of FSVA in GSM Codec with Comfort Noise Generation

In this section, we turn to another problem where the frame loss occurred in voice communication such as wireless communication (e.g. Global System for Mobile (GSM)) and network communication (e.g. Voice Over IP (VoIP)).

One cause of frame loss in these network is the the packet loss. For example, a caller uses his mobile phone to connect to a voice information service provider, such as a stock quotation server, that is connected to a traditional phone line. This phone call will be routed first to the base-station and then, connected to the traditional phone system. If the mobile phone connection between the user and the base-station incurs losses, the missing speech will be handled by the base-station, for example, by replacing it with comforting noise and the "fixed" speech is transmitted to the voice portal. The exact time of the loss information is no longer available to the voice portal server. This kind of frame loss can be regarded as a source of noise.

Our main goal in this section is to see whether FSVA with LLRT can work in such environment. We choose GSM as our target codec because of its popularity. We first give a general review of GSM environment is provided in Section 7.2.1. The next two sections, Section 7.2.2 and Section 7.2.3 then describe the two components, the speech codec and the comfort noise generator. The experiment setup and evaluation results are presented in Sections 7.2.4 and 7.2.5 respectively.

7.2.1 GSM environment

Global System for Mobile (GSM) is a second generation cellular phone system originally developed for European countries. It is the world's first cellular system to specify digital modulation and the network level architectures and services. Its specification can be found in [3] that covers all details including its service, protocol and other implementation detail. In this section, we are interested in how GSM transmission affects the quality of speech. The transmission of speech can be simplified as in Figure 7.2.

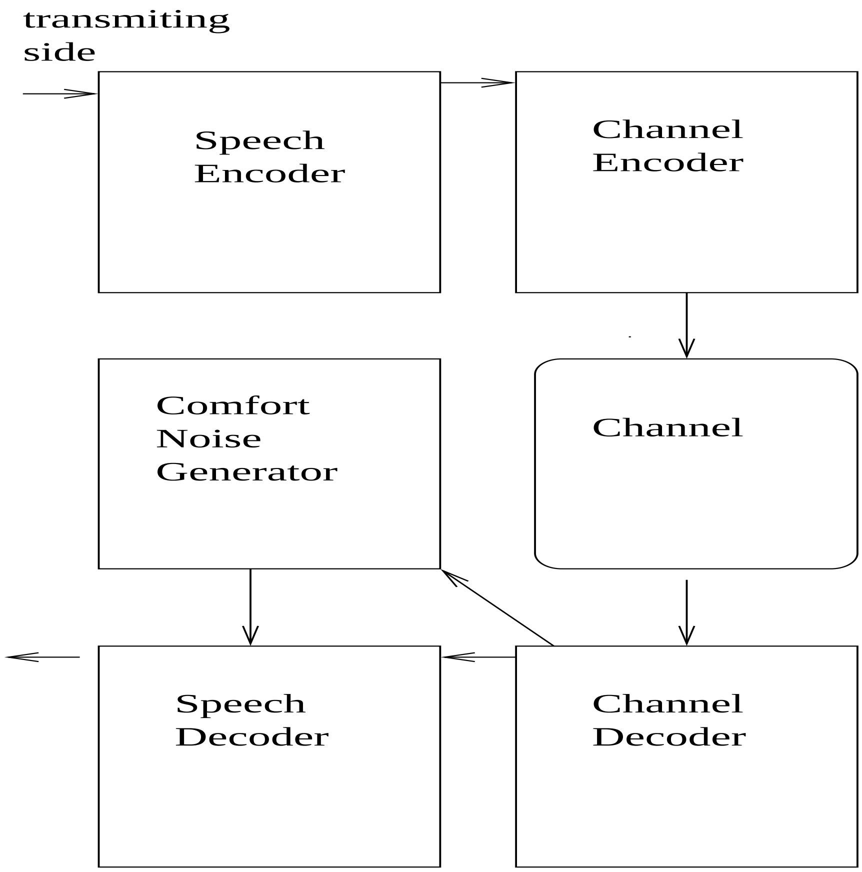

Figure 7.2: Speech Data Transmission in the GSM environment

There are two sources that can cause distortion of speech in the GSM environment. First of all, the GSM coder is based on a speech coding scheme called Regular Pulse Excitation Long Term Prediction (RPE-LTP). RPE-LTP is basically a linear prediction coding (LPC) technique that requires an error signal (or an excitation signal) to recover the speech signal in the receiver side. To reduce the bit rate of the speech encoder, the excitation signal is quantized and sent through the channel for each frame. Here, each frame is 20ms in a 8KHz system. Hence, we can expect this codec to distort the original speech signal. Another source of error is introduced by the channel. Additive noise occurred in the channel may cause some transmitted bits to be corrupted and irrecoverable during decoding.

There are two mechanisms in GSM to tackle channel corruptions. First of all, error-correcting code (ECC) is used to protect those perceptually significant bits in an RPE-LTP coded speech frame. Second, when the decoder detects that a frame is lost, it will activate the comfort noise generator (CNG) to generate comfort noise such that the whole speech signal will be less irritating to the receiver.

In this thesis, we only consider the distortion from the channel at frame loss that ignore the effect of the distortion introduced by the codec [25]. Also, we assume the rate of frame loss occurred is calculated after error correction by the channel decoder. Conceptually, the FSVA is useful in dealing with this kind of noise. However, as the comfort noise generator (CNG) may create comfort noise that is similar to the speech signal, the generated frame may be quite similar to the original speech signal. The question becomes whether the FSVA can still outperform standard Viterbi algorithm in this case. This problem is obviously related to the mechanism of the CNG. In our discussion, we will try to describe the example CNG specified in Section 7.2.3. This in turn requires our knowledge of the speech codec specified in Section 7.2.2.

7.2.2 GSM Speech Codec

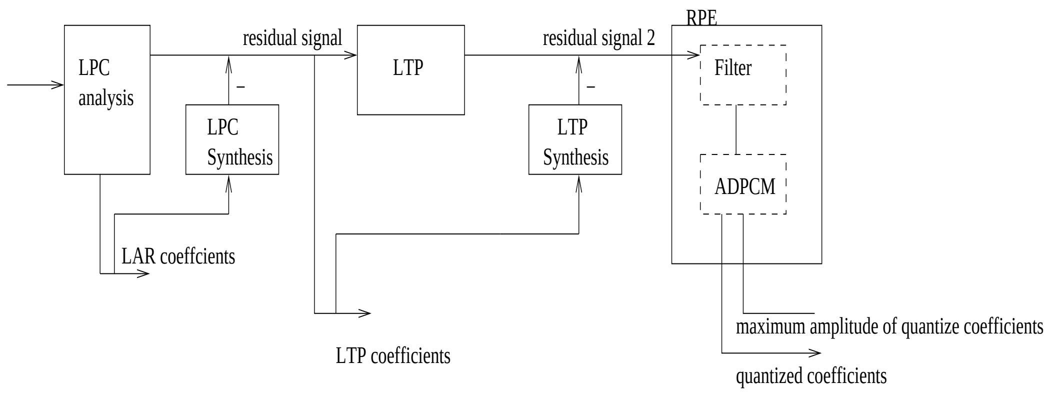

Figure 7.3: The simplified diagram of the RPE-LTP

The speech codec used in GSM is called Regular Pulse Excitation Long Term Prediction (RPE-LTP) [64]. It is a kind of linear predictive analysis and synthesis (LPAS) type codec. Its detail specification can be found in Specification GSM

06.10 [16].

We can summarize the operation of its encoder in a simple block diagram in Figure 7.3. The RPE-LTP operation includes three modules

- **Short-term analysis** or linear predictive analysis which models the vocal tract as an all pole linear filter with a transfer function described by

  $$H(z) = \frac{G}{1 + \sum_{k=1}^{M} b_k z^{-k}}$$

  Where G is the gain of the filter and z^{-1} represents a unit delay operation and the values of a_k, LPC coefficients, to characterize the speech signal. The exact relation between the excitation and the signal can be written as,

  $$s(n) = \sum_{k=1}^{M} a_k s(n-k) + e_n$$

  where e_n is the prediction error.

  The values of LPC coefficients can be found in several ways, one method called the Schur recursion which can be used to efficiently calculate a_k which minimizes the energy of prediction error. Normally, the LPC coefficients will not be coded directly, because this would require 8 to 10 bit to code them . Usually, the hyperbolic mapping of the reflection coefficients (a close relatives of LPC coefficients, please consult [49]), called log area ratio (LAR) coefficients, would be used.

  In RPE-LTP, an order 8 LPC analysis is done for every frame with 160 samples. The LPC coefficient is then transformed to the LAR coefficient. The LAR coefficients are then quantized according to its importance and using about 15% of the bits in transmission.

- **Long-term analysis**

  LPC schemes differ in the way they recreate the error signal (excitation) at the receiver. In RPE-LTP, the long-term correlation exists in the quasi periodic vibration of the vocal chords is model. This process is known as long-term prediction. A simple approximation for this process of LTP can be modeled as,

$$\frac{1}{P(z)} = \frac{1}{1 - \beta z^{-D}},$$

where β can be thought as the long term prediction coefficient and D is the predictor delay.

In RPE-LTP, the residual signal with 160 samples in each frames is subdivided into 4 sub-frames and the value of β is calculated for each sub-frame. The LTP coefficients and other information (see [64] for detail) and used up another 15 % of the bits in transmission.

- **Regular pulse excitation (RPE) Encoding**

  We can simply regard this part as a quantized version of the second residual signal produced in long term prediction process. In this process, adaptive differential pulse code modulation is used to code the residual signal for each sub-frame and the maximum amplitude of the residual signal in each sub-frame is encoded and used up the remaining 70% of the bits.

The knowledge of the GSM codec allows us to understand the example solution suggested by GSM specification of the comfort noise generator.

7.2.3 GSM Comfort Noise Generator

When frame loss occurs in the channel, what GSM 06.11 [17] based on is the first frame of identified [1] bad frames will be replaced by the previous good frame. For subsequent corrupted frames, the amplitude of the residual signal of long term prediction will be decreased after a maximum of 320ms. After 320 ms, a predefined silence frame will be used to substitute the lost frame

The above comfort noise generation can effectively reduce the problem when the frame-loss occurred in 1-frame length. However, the generated frame quality

[1] The mechanism of this identification process is not important in our discussion, we can just assume the identification is perfect in our experiment

degrades as the length of frame loss increases. In this sense, we can expect the FSVA can outperform the Viterbi algorithm when the length of frame-loss is longer than 1-frame. We will see whether this is true in our evaluation section.

7.2.4 Experiment Setup of Evaluation

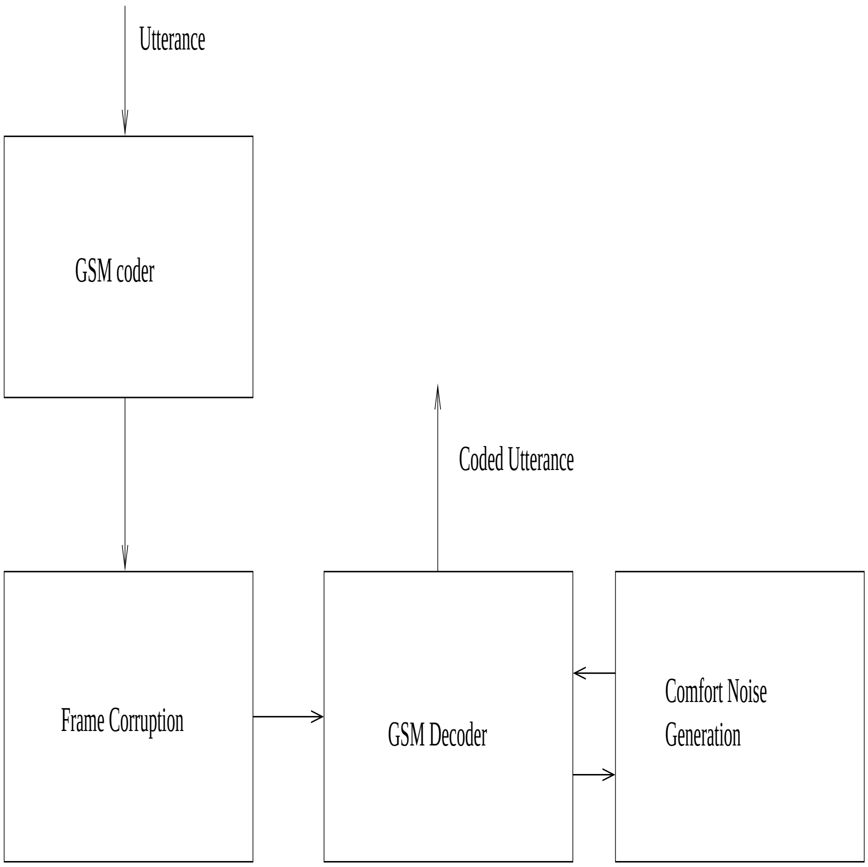

Figure 7.4: The block diagram of the experiment setup in the GSM task.

In this experiment, we try to evaluate the FSVA under frame loss with comfort noise replacement. The experiment is carried out using the AURORA database [22] which is basically a down-sampled version of TI-digits. The main reason we used it is that its sampling rate (8kHz) matches the sampling rate in GSM codec. Our baseline results are evaluated on the matched train and test environment, in which both of them are GSM coded. This matches the situation where we are able to obtain the GSM coded training utterances. The codec is publicly available in [1]. There are 8008 utterances in training. We select all clean utterances in the test set and totally there are 4004 sentences. The average baseline performance in clean coded utterances is 98.64%. The noise addition process is shown in Figure 7.4. The waveform is first transformed to the code domain. Some frames are randomly corrupted in the coded speech signal. In the decoder side, we assume the comfort noise generator have full knowledge of the positions of corruption. That means it is assume to perform optimally. The comfort noise generation is performed in the manner described in Section 7.2.3. The modified signal is decoded by the GSM decoder and results in the output corrupted utterances.

7.2.5 Experimental Results

Table 7.4 shows the results of the optimal performance of FSVA with LLRT average among the four test sets in different conditions. We adjust the duration of the frame loss and rate of the frame loss to see its effect.

D	C	BL	LLRT	threshold
1	0.01	98.03	98.12	104
2	0.01	96.47	97.40	104
3	0.01	96.10	97.19	98
1	0.02	97.71	97.95	106
1	0.05	96.31	97.20	98
1	0.1	92.98	95.33	98

Table 7.4: The performance of the the FSVA with LLRT in GSM environment using optimal threshold where L is the length of the corruption and C is the rate of corruption.

In general, we show that the FSVA with LLRT outperforms VA in all situations. However in a low rate of corruption ($C = 0.01$) and short length of corruption ($L = 1$, where each frame is 20ms long), the performance is only comparable to the baseline performance. This may be due to the fact that the

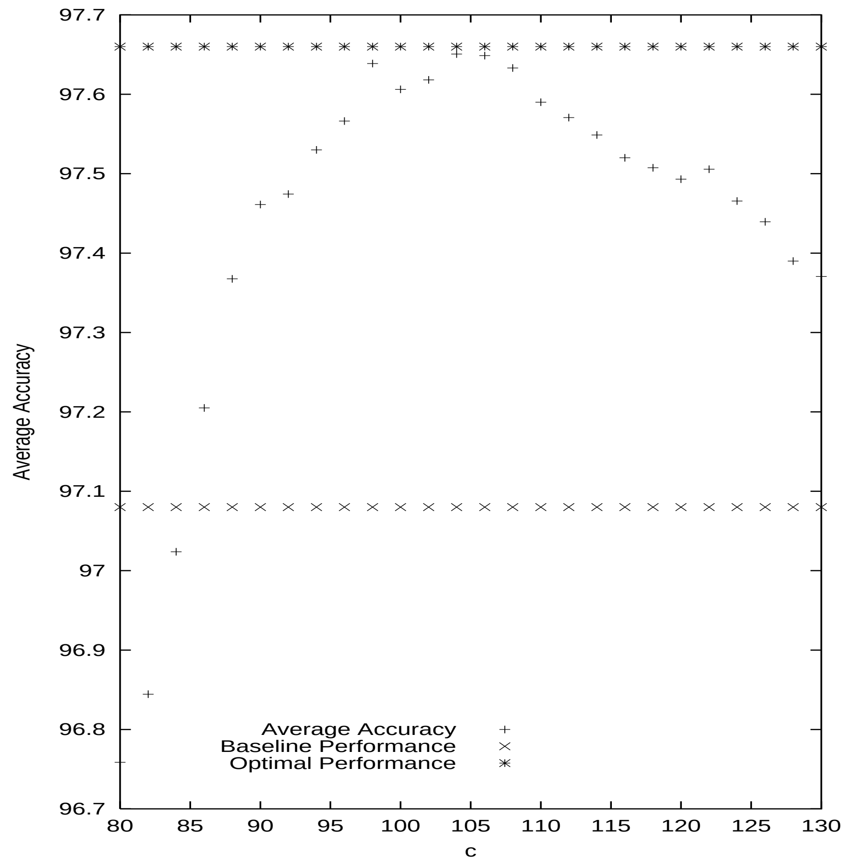

Figure 7.5: The average performance of the FSVA with LLRT against the value of $\log c$ in the GSM environment at low rate of corruption

CNG replaces the first corrupted frame by previous speech frame. In effect, this helps to keep the quality of the speech and causes little degradation. We see that in duration longer than 1, the comfort noise consists of frames which have deteriorated contents. In those cases, the FSVA skips these frames and disallows them to affect the decision. In a higher rate of corruption and with single frame corruption, the FSVA is more likely to provide gain simply because the rate of consecutive frames is being higher.

Similar to the case in short-time additive noise, we are also interested to find out whether the performance is sensitive to the value of $\log c$. In a typical corruption rate in GSM communication, the rate of corruption is assumed to be 1% and 2%. To understand the performance of FSVA in this situation, we present the average performance against the value $\log c$ in Figure 7.5. Similar to what was obtained in Sec. 7.1, we see that the FSVA is useful in a wide range of values and there is range of value such that a single value of the threshold can be suitable for all conditions and have similar performance as the use of the optimal value of the parameter.

7.3 Summary and Conclusions

In this chapter, we have evaluated the FSVA with LLRT in three tasks. These include short-time noises that are additive and replaced in nature. This completes the evaluation of FSVA of this thesis.

We showed that the FSVA with LLRT performs satisfactorily in additive short-time noise with different amplitudes, rates of corruption and lengths of corruption. The method works especially well when SNR is smaller (or the gain of noise large). However, under all conditions, it never degrades. Moreover, the performance of the FSVA with LLRT is insensitive to the LLRT threshold and a single threshold can be used such that the FSVA with LLRT performs well in all conditions.

In the case of network transmission of voice, even under the complication of a comfort noise generator. We got similar conclusion as in the case for additive noise.

Chapter 8

Conclusions and Future Work

8.1 Contributions

In this thesis, we investigated the mismatch problem in speech recognition that caused by short-time noise or partial and temporal corruption (PTC). By short-time noise or PTC, we mean those time-limited noises with duration shorter than the speech signal and with random amplitudes, spectral contents, characteristics of occurrence and random durations time.

We introduced an algorithm called the frame-skipping Viterbi algorithm (FSVA). The algorithm is based on the missing feature theory (MFT) that suggests that those corrupted or missed data should be omitted. We proposed two possible implementations for it. While the topological-space expansion approach can be implemented using standard speech recognition softwares, the state-space expansion approach is more efficient in terms of computation and memory.

One important parameter in the FSVA framework is the number of skips. To make the algorithm useful across different noise environments, it is important that the number of skips can be determined automatically. We proposed the use of log likelihood ratio thresholding (LLRT) to control the number of skips for every sentence. This is an algorithm which can automatically determine the number of skips without the knowledge of the noise environment by setting a single threshold. This threshold is shown to be insensitive to the noise environment. The development of this technique allows the FSVA to handle short-time noise with multiple conditions.

We have tested the usefulness of the FSVA with LLRT in different noise environments, including that of Gaussian replacement noise, PTC addition noise with different SNRs, durations and rates of corruption as well as frame-loss in the GSM environment. The FSVA with LLRT is shown to perform equally or better than the standard Viterbi algorithm when $SNR \leq 0$.

One drawback of the FSVA with LLRT framework is the increase in the memory usage. We also proposed a simplified version the FSVA that used the same amount of memory as the standard Viterbi algorithm. We called it the simplified FSVA (SFSVA). The SFSVA incorporated the LLRT into the FSVA. In our preliminary study using Gaussian replacement noise, the SFSVA were shown to give comparable performance as the FSVA with LLRT.

There is another idea called the extended union model[42] that shares the same problem formulation to our work. In it original work, the model is used to tackle noise that appears in segment-level. In other words, even the corruption appears in only a couple of frames, the whole segment is regarded as corrupted. We found that by modifying the formulation of the FSVA, we can apply the EUM on frame-level. We found that the search technique in the FSVA can be generalized into an efficient search algorithm for EUM at frame level. We called it the frame-based EUM Viterbi algorithm (FEVA).

In conclusion , the FSVA and its various variants are shown to be a useful tool to deal with partial and temporal corruptions. Though our evaluation only focuses on artificially created conditions, we believed that it can be used in real-life situations.

8.2 Future Work

There are several possible future works for this thesis.

The development of our algorithm is guided from Missing Feature Theory (MFT). As suggested by MFT, not only one can omit the contribution of the corrupted components in the decision process, we can also try to reconstruct their values using those clean components. This is called imputation in general. It will be interesting to see whether we can develop robust algorithms based on this idea. This idea have previously been applied to spectrogram reconstruction and band reconstruction [13].

Back to the algorithm itself, as we know that the determination of number of skips is the most crucial problem and what we have proposed in this thesis is just one possible method, the log likelihood ratio thresholding (LLRT) to decide the number of skips. If we compare the result selected by the LLRT with the oracle results obtained from K skips, we see that there is still a large room for improvement. We believe the development of new skip determination algorithm is another interesting work.

Another interesting point we observe in this work is that the search of state sequence using robust likelihood is found to be approximately equivalent to search of state sequence with maximum probability of extended union model (EUM). This fact suggests that if it may exist a class of functions which are similar to robust likelihood and probability of extended union model (EUM) which allows the dynamic algorithm to be applied on them. In general, we believe the searching and identification of this class of function is interesting because it may allow us to construct criterion function for the Viterbi algorithm which can be searched dynamically.

In terms of evaluation, our evaluation in replacement noise in voice communication in network can be extended to testing in different network environments (For example, extend the codecs to VoIP and third generation mobile communication systems) and different comfort noise generation schemes. (For example, extend to other more complicated methods of reconstruction of the signal.)

Finally, as Viterbi algorithm and trellis are the basis of techniques in many fields such as trellis code modulation, convolutional decoding and image processing. We believe that similar techniques as we developed in this thesis can be applied in these fields to deal with burst noise. For example, the use of FSVA in convolutional decoding may allow identification of incorrect code segments in the decoding module. How we can create a low complexity implementation of such decoding algorithm will be an interesting future direction.

8.3 Published Papers

During this research, we have published 3 papers,

- "Pruning of State-Tying Tree Using Bayesian Information Criterion" [10]
- "Robust Speech Recognition against Packet Loss" [59]
- "Robust Speech Recognition against Short-Time Noise" [60]

Appendix A

Union Probability

We first describe the basic probability rule of union events.

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B) \tag{A.1}$$

In general, union of events $(b_1, \ldots, b_N)$ can be computed recursively based on the following equation if we assume event independence.

$$\begin{aligned}
& P(b_1 \vee \ldots b_N) \\
= \;& P((b_1 \vee \ldots b_{N-1}) \vee b_N) \\
= \;& P(b_1 \vee \ldots b_{n-1}) + P(b_N) \\
- \;& (b_1 \vee \ldots b_{n-1}) P(b_N) \\
\approx \;& P(b_1 \vee b_2 \ldots \vee b_{n-1}) + P(b_N)
\end{aligned}$$

Notice that in practice the cross product term usually contributes little to the union probability. Thus, we can omit and the following approximation as in the last approximation.

Appendix B

Proof of the EUM Formula

This section presents the proof of the EUM formula (Equation 4.16) in Section 4.5.3. Using the definition of EUM in Section 3.3.3. EUM of t-frame given a state sequence can be expressed as,

$$P(\mathring{O}^M_{\vee(t)}) \quad = \quad P(\bigvee_{i=0}^{C^t_M} l_i),$$

where l_i is the i-th pattern of corruption. We can divide the C^T_M patterns into two subsets. The first set, L_1, includes those cases with o_N has omitted in the joint probability, the second set, L_2, does not.

Hence, Equation B.1 can be rewritten as

$$\begin{aligned} P(\mathring{O}^M_{\vee(t)}) &= P(L_1 \vee L_2) \\ &- P(L_1) + P(L_2) - P(L_1 \wedge L_2). \end{aligned}$$

For L_1, since o_t has been omitted, there are only $M - 1$ corruptions occurred in previous $t - 1$ frames. Thus, one can express this event as $\mathring{O}^M_{\vee(t-1)}$. i.e. $P(L_1) = P(\mathring{O}^M_{\vee(t-1)})$

For L_2, since o_t has not been omitted, there must be M corruptions occurred in previous $t - 1$ frames. Thus one can express this $(\mathring{O}^{M-1}_{\vee(t-1)} \wedge o_t)$.

If we assume independency between frames, the above discussion suggests that Equation B.1 can be rewritten as

$$
\begin{align}
P(\mathring{O}^{M}_{\vee(t)}) &= P(\mathring{O}^{M}_{\vee(t-1)}) + P(\mathring{O}^{M-1}_{\vee(t-1)} \wedge o_t) \nonumber \\
&- P(\mathring{O}^{M}_{\vee(t-1)} \wedge (\mathring{O}^{M-1}_{\vee(t-1)} \wedge o_t)) \nonumber \\
&= P(\mathring{O}^{M}_{\vee(t-1)}) + P(\mathring{O}^{M-1}_{\vee(t-1)})P(o_t) \tag{B.1} \\
&- P(\mathring{O}^{M}_{\vee(t-1)} \wedge P(\mathring{O}^{M-1}_{\vee(t-1)})P(o_t)) \nonumber \\
&\approx P(\mathring{O}^{M}_{\vee(t-1)}) + P(\mathring{O}^{M-1}_{\vee(t-1)})P(o_t) \tag{B.2}
\end{align}
$$

The last approximation omits the contribution of the cross-product term that is usually small in speech recognition.

Appendix C

Extending Topological Space Approach to Continuous Speech Recognition

This is intrinsically difficult to extend topological space approach to continuous speech recognition. In the general decoder structure that is illustrated in Figure C.1.

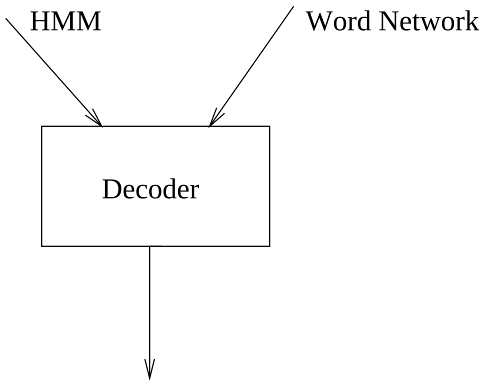

Figure C.1: A block diagram for a generic decoder

We now illustrate the difficulty of incorporating frame-skipping in topological space extension using a simple example. Suppose we are working on connected digit recognition task, then the word network can be expressed as figure C.2. If we substitute the FSHMM with 1 frame skip to the HMM without changing the word network, it will be impossible for us to control the number of skips in the decoding because we allow at least one frame skipped in every FSHMM. This is actually dangerous because, the number of skips may mismatch for a great deal with the actual situation.

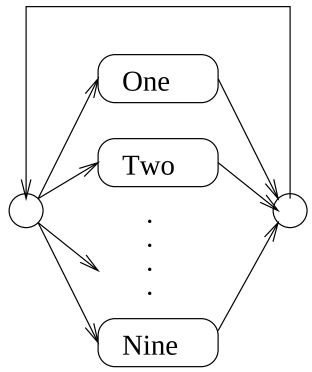

Figure C.2: The word network for connected digit recognition.

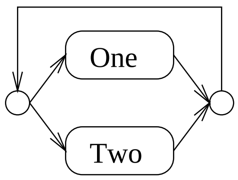

Figure C.3: A simplified network for connected digit recognition.

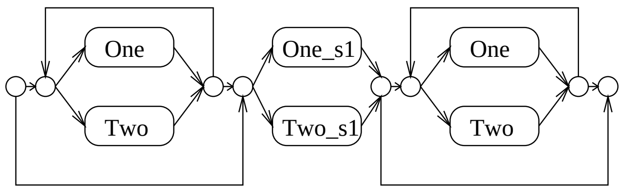

Figure C.4: The word network that skips one frame.

If we really want to constrain the number of frames being skipped, we will avoid the above situation by using following trick in word network manipulation. The idea is that we need to constraint the network such that a skipped HMM can be only applied once. Consider a simplified version of no-grammar network in figure C.3. A frame-skipped word network (FSWN) can be found as figure C.4. The networks will always permit the one skipping in a sentence. Similarly, if we want to FSWN with two skipped, we can constrain the network such that either a model can skip two frames or two models skipping two frames. This is shown in figure C.4.

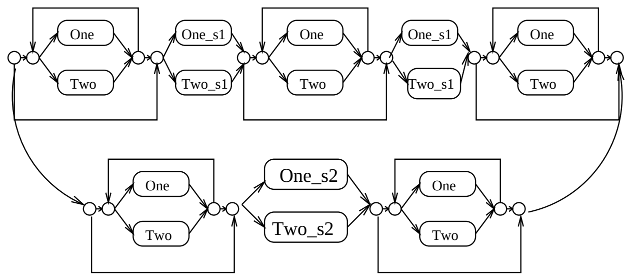

Figure C.5: The word network that skips two frames.

In this example, it is also possible to work out the network analytically by expressing the word network as extended regular expression grammar. Define G_k as the network that skip for k frames. S_k as a network which give a choice for passing words that skips k frames. Using this notation, G_0 will represent a left-to-right network as shown in Figure C.3.

As we can see, for network that can skip for 1 frame, the network will be

$$N_1 = N_0 S_1 N_O \tag{C.1}$$

and the network for skipping 2 frames will be

$$N_2 = N_0 S_1 N_0 S_1 N_0 | N_0 S_2 N_O = N_0 (S_1 N_O S_1 | S_2) N_O \tag{C.2}$$

From the factorization, we see that we can always separate the skipping network to initial and final words. We denote the term in the bracket as C_k. The intuitive meaning of this is a network that must skip at the beginning and the ending word. Using some very simple reasoning. We can write down the recursive formula for C_k

$$C_k = S_1 N_0 C_{k-1} | S_2 N_0 C_{k-2} | \dots | S_{k-1} N_0 C_1 | S_k \tag{C.3}$$

and the expression for N_k would be

$$N_k = N_0 C_k N_0 \tag{C.4}$$

This algorithm is advantageous because we can apply some well-developed speech recognition research tools such as HTK to implement FSVA. However, it is obvious that this algorithm is very memory-intensive.

Appendix D

Setting of Speech Parametization

The following is the exact setting of feature parametization. This setting is in the format of config file of HTK(V2.2 or above) HCopy command.

SOURCEKIND = WAVEFORM

SOURCEFORMAT = NIST

SOURCERATE = 500

ZMEANSOURCE = FALSE

TARGETKIND = MFCC_E_D_A

TARGETRATE = 100000

SAVECOMPRESSED = FALSE

WINDOWSIZE = 250000.0

USEHAMMING = TRUE

CEPLIFTER = 22

NUMCHANS = 26

NUMCEPS = 12

ENORMALISE = TRUE

ESCALE = 1.0

HPARM : SAVEWITHCRC = F

Appendix E

Basic Setting in Noise Addition

In 8668 TI-digits utterances spoken by adult, if we divide the waveform by 25ms without overlapping of frames. We will get 606846 frames. We then corrupt 10% of frames. I.e., about 60684 frames are corrupted. Note that this includes frames in non-speech words just like silence and short-pause we have defined in Section 5.1.1.

Appendix F

Waveforms and Spectrograms of Partially Temporal Corrupted Noises

In this section, all the waveforms and spectrograms of the noise will be created. There are totally 7 different waveforms has been used in the PTC experiment.

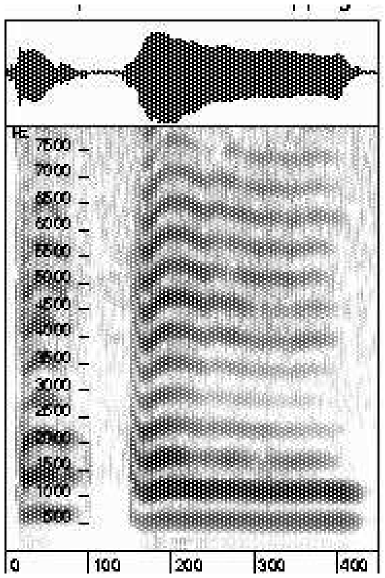

Figure F.1: The waveform and the spectrogram of ICQ message acknowledgment

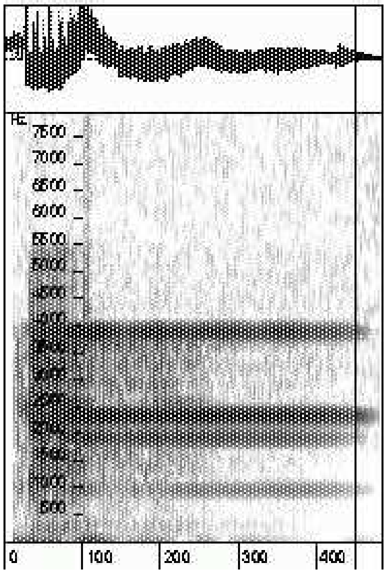

Figure F.2: The waveform and the spectrogram of ICQ message bell

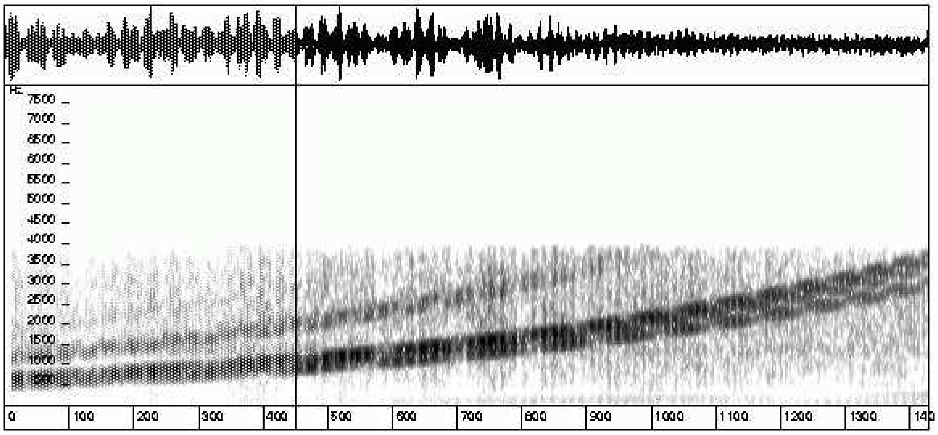

Figure F.3: The waveform and the spectrogram of the connect tone

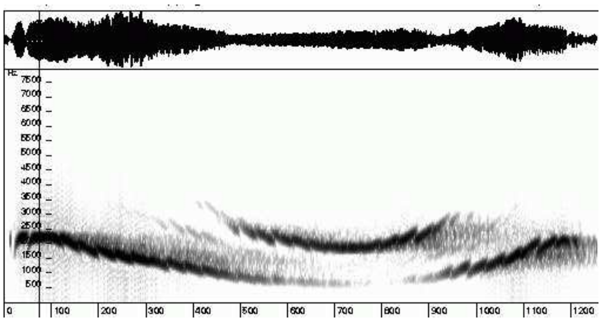

Figure F.4: The waveform and the spectrogram of the contact tone

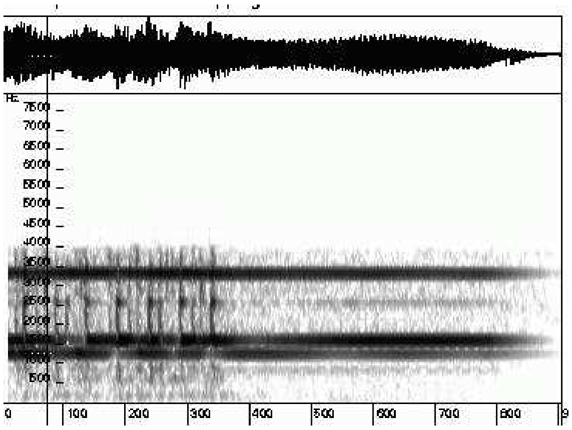

Figure F.5: The waveform and the spectrogram of the ring-in tone

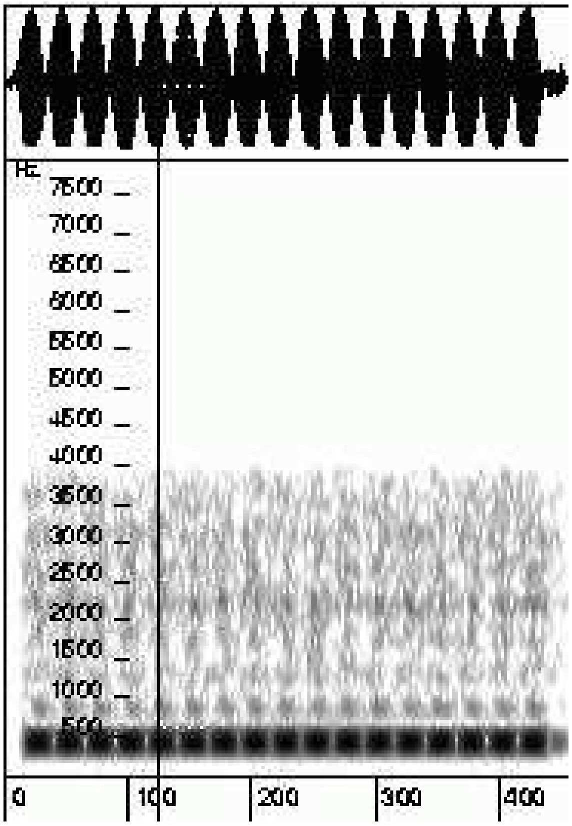

Figure F.6: The waveform and the spectrogram of the ring out tone

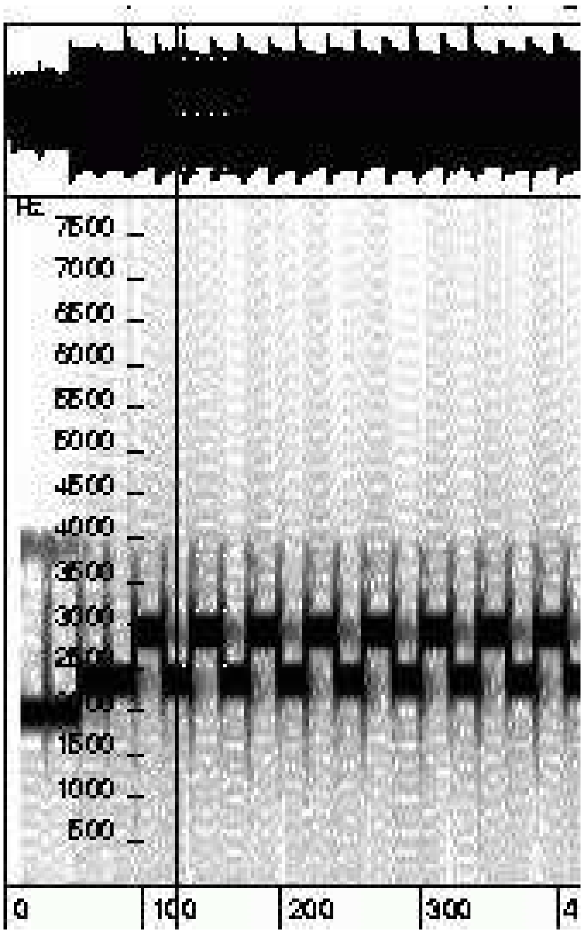

Figure F.7: The waveform and the spectrogram of the whistle bells

Bibliography

[1] Available at ftp://ftp.cs.cmu.edu/project/fgdata/speech-compression/GSM/GSM-1.0.tar

[2] Available at http://www.dspguru.com/howto/tech/wgn.htm.

[3] Available at http://www.etsi.org/

[4] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, eds. 1992, Norwell, MA, Kluwer Academic Publishers.

[5] S. Ahmed, V. Tresp. "Some Solutions to the Missing Feature Problem in Vision," in *Advances in Neural Information Processing System* 5. 1993.

[6] J. Bellegarda, "Statistical Techniques for Robust ASR: Review and Perspective,", *Eurospeech*, 1997.

[7] H. A. Bourland, N. Morgan, *Connectionist Speech Recognition*, eds. 1994, Norwell, MA, Kluwer Academic Publishers.

[8] R. A. Brualdi, *Introductory Combinatorics*, 3rd Edition, Engelwood Cliffs, New Jersey, Prentice Hall Inc.

[9] H. Bunke, T. Caelli, "Hidden Markov Models : Applications in Computer Vision," *Series in Machine Perception and Artificial Intelligence*, Vol. 45. World Scientific.

[10] Y. C. Chan, M. Siu, "Pruning of State-Tying Tree Using Bayesian Information Criterion,"*Int. Conf. on Speech and Language Processing*, Beijing, 2000.

[11] E. Chang and R. Lippmann., Improving Word-spotting Performance with Artificially Generated Data, *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1993, pp.258-261.

[12] S. S. Chen, E. M. Eide, M. J. F. Gales, R. A. Gopinath , D. Kanevsky, P. Olsen, "Recent Improvement to IBM's Speech Recognition System for Automatic Transcription of Broadcast news," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1999.

[13] M. Cooke, P. Green, L. Josifowski, A . Vizinho Robust ASR with unreliable data and Minimal assumptions, Robust 99

[14] S. Das, R. Bakis, A. Nadas, D. Nahamoo, M. Pichemy, "Influence of Background Noise and Microphone on the Performance of the IBM TANGORA Speech Recognition System," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1993, pp.258-261.

[15] S. B. Davis, P. Mermelstein, "Comparison of Parametric Representation for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. of Signal and Speech Processing*, 1980, **28**(4) pp.357-366.

[16] Digital cellular telecommunications system (Phase 2); Full Rate Speech ; Part 2 : Transcoding (GSM 06.10 version 4.1.1). Available at http://www.etsi.org/

[17] Digital Cellular Telecommunications System (Phase 2+); Full Rate Speech; Substitution and Muting of Lost Frames for Full Rate Speech Channels. Available at http://www.etsi.org/

[18] Y. Ephraim and L. Rabiner, "One the Relations Between Modeling Approaches for Speech Recognition," *IEEE Trans. Information Theory*, **36**(2),pp 372-380, 1990.

[19] M. J. F. Gales, "Model-based Techniques for Noise Robust Speech Recognition", Ph. D Thesis in Engineering Department, Gouville and Caius college, September, 1995, Cambridge.

[20] Glossary of Institute for Telecommunication Science http://www.its.bldrdoc.gov/fs-1037/dir-018/_2685.htm"

[21] J. H. L. Hansen, L. M. Arslan, "Robust Feature Estimation and Objective Quality Assessment for Noisy Speech Recognition Using the Credit Card Corpus,",*IEEE Trans. Speech and Audio Processing*, 3, 1995, pp.169-194.

[22] H. G. Hirsch, D. Pearse, "The AURORA Experimental Framework for the Performance Evaluation of Speech Recognitions Systems under Noisy Conditions," *ASR 2000 Automatic Speech Recognition: Challenges for the Next Millennium*, 2000.

[23] X. D. Huang, A. Acero, H. W. Hon, *Spoken Language Processing : A Guide to Theory, Algorithm and System Development*, Upper Saddle River, New Jersey, Prentice Hall Inc., 2001

[24] X. D. Huang, Y. Ariki, M. A. Jack, *Hidden Markov Model for Speech Recognition*, 22 George Square, Edinburgh University Press, 1990.

[25] J. M. Huerta, Speech Recognition in Mobile Environments, PhD thesis CMU April,2000.

[26] F. Jelinek, *Statistical Methods for Speech Recognition* The MIT Press, 1997.

[27] D. V. Johan, F. D. Wet, B. Cranen, L. Boves, Missing Feature Theory in ASR: Making Sure Miss the Right Type of Features, *Proceeding. Workshop on Robust Methods in Adverse Conditions*, Tampere, Finland, 1999

[28] B. H. Juang, W. Chou, C. H. Lee, "Statistical and Discriminative Methods for Speech Recognition" in *Automatic Speech and Speaker Recognition'*, C. H. Lee, F. K. Soong, K. K. Paliwal, eds. 1994, Norwell, MA, Kluwer Academic Publishers.

[29] J. Junqua and Y. Anglade, "Acoustic and Perceptual Studies of Lombard Speech : Application to Isolated Word Automatic Speech Recognition," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1990, pp. 841-844.

[30] M. Kadirkamanathan, A. P. Varga, "Simultaneous Model Re-estimation From Contaminated Data by "Composed Hidden Markov Modeling"," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1991.

[31] S. A. Kassam and H. V. Poor, "Robust Techniques for Signal Processing: A Survey,", *Proceedings. of IEEE*, **73**(3), 1985, pp.433-481.

[32] C. Kermorvant, A. Morris, A Comparison of Two Strategies for ASR in Additive Noise: Missing Data and Spectral Subtraction. ICASSP 99.

[33] C. H. Lee, F. K. Soong, K. K. Paliwal, *Automatic Speech and Speaker Recognition*, eds. 1996, Norwell, MA, Kluwer Academic Publishers.

[34] R. G. Leonard, "A Database for Speaker-Independent Digit Recognition," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1984.

[35] R. P. Lippmann, "Review of Neural Network for Speech Recognition" in *Readings in Speech Recognition*, A. Waibel, K. F. Lee, Morgan Kaufmann, San Mateo, CA, 1990.

[36] R. Lippmann, B. Carlson, "Robust Speech Recognition with Time-varying Filtering Interruptions and Noise," *Proceedings of the Automatic Speech Recognition Workshop*, 1997.

[37] R. Lippmann, B. Carlson, "Using Missing Feature Theory to Actively Select Features for Robust Speech Recognition With Interruptions, Filtering and Noise," Eurospeech, 1997, pp.37-40.

[38] R. J. A. Little, D. B. Rubin. *Statistical analysis with Missing Data*, Wiley series in probability and mathematical statics. 1987.

[39] M. Matassoni, M. Omologo, D. Giuliani, "Hands-Free Speech Recognition Using a Filtered Clean Corpus and Incremental HMM Adaptation," *Int. Conf. of Spoken Language Processing*, 2000, pp.1407-1410.

[40] M. Merhav, C. H. Lee, "A Minimax Classification Approach with Application to Robust Speech Recognition" *IEEE Trans. on Speech and Audio Processing*, 1993, **1**(1), pp.90-100.

[41] J. Ming. available at http://www.cs.qub.ac.uk/ J.Ming/Html/Robust.hml".

[42] J. Ming, F. J. Smith. "Union: A Model for Partial Temporal Corruption of Speech," *Computer Speech and Language*, 2001.

[43] J. Ming, F. J. Smith. "Union: A New Approach for Combining Sub-Band Observations for Noisy Speech Recognition," *IEEE Workshop on Robust Speech*, Finland, 1999.

[44] B. B. J. Moore, *An Introduction to the Psychology of Hearing 4th Edition.* Academic Press.

[45] P. J. Moreno, "Speech Recognition in Noisy Environment," Ph. D Thesis, ECE Department, May 1996, Carnegie Mellon University.

[46] M. Mouly, M. B. Pautet, *The GSM System for Mobile Communications, a Comprehensive overview of the European Digital Cellular Systems*, 1992.

[47] A. Nadas, D. Nahamoo, M. Pichemy, "Speech Recognition Using Noise-Adaptive Prototypes," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1998, pp.517-520.

[48] A. Nean, M. Hayes, "Face recognition using and embedded HMM", *IEEE Int. Conf. on Audio and Video-based Biometric Person Authentication*, Washington D.C. 1999.

[49] A. V. Oppenheim, R. W. Schafer, *Discrete-time Signal Processing*, eds. 1989, Engelwood Cliffs, New Jersey, Prentice Hall Inc..

[50] A. V. Oppenheim, R. W. Schafer, *Discrete-time Signal Processing*, eds. 1989, Engelwood Cliffs, New Jersey, Prentice Hall Inc..

[51] J. E. Porter, S. F. Boll. “Optimal Estimators for Spectral Restoration of Noisy Speech,” *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1984.

[52] L. Rabiner, B. H. Juang, *Fundamentals of Speech Recognition*, eds. 1993, Engelwood Cliffs, New Jersey, Prentice Hall Inc..

[53] M. G. Rahim, B. H. Juang, “Signal Bias Removal by Maximum Likelihood Estimation for Robust Telephone Speech Recognition,” *IEEE Trans. On Speech and Audio Processing*, 1996.

[54] T. S. Rappaport, *Wireless Communications, principles and practice* , Prentice Hall, 1998.

[55] S. M. Ross, Introduction to Probability Models, 5-th Edition, Academic Press.

[56] D. B. Rubin, J. L. Schafer “Multiple Imputation for Missing Data Problem,” in *Continual Education Course Prepared for Joint Statistical Meetings*, 12 Aug, 1998.

[57] J.F. Schafer, “Analysis of Incomplete Multivariate Data,” Chapman and Hall, London.

[58] R. Schwartz and Y. L. Chow, “The N-best Algorithm: an Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses,”*IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1990, Albuquerque, pp. 81-84.

[59] M. Siu, Y. C. Chan, “Robust Speech Recognition against Packet Loss,”*Eurospeech*, Scandalvania, 2001

[60] M. Siu, Y. C. Chan, “Robust Speech Recognition against Short-Time Noise,”*Int. Conf. on Speech and Language Processing*, 2002, to be appeared.

[61] W. Strange, J.J. Jenkins, T.L. Johnson, “Dynamic Specification of coarticulated vowels,” *Journal of the Acoustical Society of America*, **74**(3) pp.695-705.

[62] H. Van Trees, *Detection, Estimation and Modulation Theory: Part 1* New York, Wiley pp.86-96.

[63] A. P. Varga, R. K. Moore, "Hidden Markov Model Decomposition of speech and noise," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, April, 1990, Albuquerque.

[64] P. Vary, K. Hellwig, R. Hofmann, R. J. Sluyter, C. Galand and M. Rosso, "Speech Codec for the European Mobile Radio System," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1988, pp.227-230.

[65] A. Waibel, K. F. Lee, Readings in Speech Recognition, San Mateo, CA, Morgan Kaufmann Publishers Inc. 1990.

[66] P. C. Woodland, J. J. Odell, V. Valtchev, S. J. Young. "Large Vocabulary Continuous Speech Recognition Using HTK," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1994.

[67] S. J. Young, J. J. Odell, D. Olasen, V. Valtchev, P. C. Woodland, *The HTK Book (for HTK Version 3.0)*, 1999.

[68] S. J. Young, J. J. Odell and P. C. Woodland, "Tree-Based State Tying for High Accuracy Acoustic Modelling," *ARPA Workshop on Human Language Technology*, 1994, pp 286-291.

[69] S. J. Young, P. C. Woodland, "Clustering in Hidden Markov Model-Based Continuous Speech Recognition," *Computer Speech and Language*, 9, pp. 368-383.