5-2009

Investigation of Training Algorithms for Hidden Markov Models Applied to Automatic Speech Recognition

Eric Fang
Clemson University, efang@clemson.edu

Investigation of Training Algorithms for Hidden Markov Models Applied to Automatic Speech Recognition

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Eric Fang
May 2009

Accepted by:
Dr. John Gowdy, Committee Chair
Dr. Robert Schalkoff
Dr. Stanley Birchfield

ABSTRACT

The work presented in this thesis focuses on simulating a speech recognizer which is trained by different people with different speaking styles and investigates how sensitive the training and recognition processes are to the variations in the training data. There are four main parts to this work. The first involves an experiment of weighting methods for training with multiple observation sequences. The second involves the testing of different initial parameters. The third part includes the first experiment involving training with multiple observation sequences. The model's sensitivity to variations in training data was evaluated by comparing the cases of different values of ε. The final part varied the observation vectors with the variation restricted to only one of the eight positions in the sequence. The experiment was repeated for each of eight positions in the observation sequence, and the effect on recognition was evaluated.

ACKNOWLEDGMENTS

I am very thankful to my advisor, Dr. Gowdy, and also to the other members of my committee: Dr. Robert Schalkoff and Dr. Stanley Birchfield. They and many other professors have given knowledge and encouragement during my time at Clemson University. I would also like to thank the graduate students in Speech Analysis Laboratory for their friendship and support. Special thanks goes to the Engineering and Science Education Department for supporting me financially for this degree.

TABLE OF CONTENTS

Table of Contents (Continued)

Page

LIST OF TABLES

List of Tables (Continued)

Table Page

LIST OF FIGURES

Figure Page

CHAPTER ONE

INTRODUCTION

Since speech is the most efficient way to exchange information for most people, speech recognition has been an important research topic in the last few decades. The goal of speech recognition is to create machines which can receive spoken words and to recognize them. Generally, an environment without noise is required for accurate recognition. However, noise usually occurs in houses, supermarkets, vehicles or other locations where speech recognition might be useful. Furthermore, it is also necessary to recognize the spoken words from people with different speaking styles. To increase the probability of identifying the correct words or phonemes, speech recognition has become an important research area.

The understanding of speech recognition has increased at a remarkable rate and has been implemented in various forms [1], including dynamic time warping (DTW), hidden Markov model (HMM), language modeling and artificial neural networks (ANNs). First of all, DTW [2] is a feature-matching scheme that accomplishes "time alignment" of the sets of test features and the sets of reference through a dynamic programming (DP) procedure. In other words, it is a dynamic programming method for extending or compressing observations to account for variations in length of time of phonemes or a spoken word. Secondly, HMM [2] is a "stochastic finite state automation" which can be used to model speech utterances. The probability of the observation sequence being produced over the model states are summed and compared for the maximum likelihood of a word or phoneme. In contrast, a second class of stochastic techniques based on the

ANN involve exploring an alternative computing architecture. Finally, language modeling is concerned with the recognition of large sentences by decomposing them into words according to rules that reduce entropy. Since a speech signal can be viewed as a short-time stationary signal or a piecewise stationary signal and can be trained automatically by using Hidden Markov Model, modern speech recognition systems are generally based on HMMs.

1.1 Overview of Hidden Markov Model

Hidden Markov Models are stochastic models which were first studied in the late 1960s and early 1970s have become a successful machine learning technique for speech recognition [3]. There are several reasons why the model became so popular. Firstly, the models are very rich in mathematical structure and can form the theoretical basis for use in a wide range of applications. In addition, the model can be trained automatically. Moreover, the models work very well in practice for several important applications when they applied properly.

The state is directly visible to the observer in the regular Markov Model, therefore, the state transition probabilities are the only parameters. In contrast, although the variables influenced by the states are visible, the states are not directly visible in the Hidden Markov Model. The sequence of tokens generated by the HMM gives some information about the sequence of states since each state has a probability distribution over the possible output tokens.

Furthermore, Hidden Markov models are especially known for their applications to speech recognition, handwriting recognition and gesture recognition.

1.2 Motivation of Hidden Markov Model

Three kinds of problems are associated with the Hidden Markov Model [2]. First of all, given the parameters of the model m , compute the probability $P(\mathbf{O} \mid m)$ that an observation sequence $\mathbf{O}$ is produced, given the model m . The problem can be solved by the forward-backward algorithm [13]. This is called the "any path" method. Secondly, given the parameters of the model then find the most likely sequence of hidden states that could have generated a given output sequence by using the Viterbi algorithm [14]. This is called the "best path" method. Finally, given an output sequence and find the most likely set of state transition and output probabilities. In other words, the issue here is to train a particular HMM to correctly represent its designated word. The training problem, which can be solved by the Baum-Welch re-estimation algorithm, is a key aspect of speech recognition.

One factor of interest for the HMM training is the choice of initial parameters [4]. Although the training procedure is guaranteed to reach a critical point of $P(\mathbf{O} \mid m)$, it is typically a local maximum. Therefore, different starting values of matrices A and B could yield models with higher or lower values of $P(\mathbf{O} \mid m)$. Besides, particle swarm optimization (PSO) [5] and genetic algorithm (GA) [6] have been developed to estimate optimal parameters of HMM. In order to improve system performance, finding the global maximum has become a focus of research.

1.3 Overview of Thesis

In order to provide a more complete representation of the statistical variations likely to be present across utterances, it is necessary to train a given Hidden Markov Model with multiple training utterances [2]. The next chapter presents background material for the HMM. After the background discussion, the main work of the thesis is to simulate a speech recognizer which is trained by different people with different speaking styles. In addition, this thesis investigates how sensitive the training and recognition processes are to the variations in the training data. The main work of this thesis is presented in Chapters 3 and 4, showing how sensitive the training model is. The first part discusses an experiment on training with multiple observation sequences and how to choose the initial parameters of the model. After that, the sensitivity of training to variations in the training data is determined by comparing the case of different values of ε. Finally, a summary discussion concludes the thesis.

CHAPTER TWO

HIDDEN MARKOV MODEL FOR SPEECH RECOGNITION

The hidden Markov model is a statistical model with unknown parameters. Furthermore, the challenge is to determine the hidden parameters from the observable data. Since the Markov chain was first constructed by a Russian scientist in the early 1900s [7], it has become the most successful tool for speech recognition. Also, the extracted model parameters can be used to perform other analysis such as pattern recognition, handwriting recognition and gesture recognition.

2.1 History and Development

Hidden Markov Models were first described in a series of statistical papers by L.E. Baum and other authors in the 1960s [8]. The model precedes its use in speech processing and became widely used and known in the speech field starting in the mid-1970s [9]. Baker at Carnegie-Mellon University [10] and Jelinek and colleagues at IBM [11] are generally known as the first researchers to apply Hidden Markov Models to speech recognition. Similar work on the HMM was also developed at the Institute for Defense Analysis in the 1970s [12]. Finally, HMMs have become the most successful technique in speech recognition after the pioneering work in the 1970s and 1980s. Also, HMMs began to be applied to the analysis of biological sequences such as DNA in the second half of the 1980s. This chapter details the definition and assumptions of HMMs, how they are trained, and some other practical issues.

2.2 Definition of the Hidden Markov Model

The hidden Markov model, which is an extension of the Markov chain is a double-embedded stochastic process [13]. Since it models an observable stochastic process with a hidden stochastic process, it is called a doubly stochastic process.

The model is usually defined as a parameter set {S, A, B, π(1), O}. [14] [26]

- **States** (S): S is the total number of states in the model that represents the state space.
- **Transition probabilities** (A): A is a matrix specifies the transition probabilities between states. A(i|j) represents the probability of transitioning from state j to state i. The state transition probabilities are assumed to be stationary in time so that A(i|j) does not depend upon the time when the transition occurs.

$$A = \begin{pmatrix} a(1|1) & a(1|2) & \cdots & a(1|S-1) & a(1|S) \\ & \ddots & & & \\ & & a(i|j) & & \\ & & & \ddots & \\ a(S|1) & a(S|2) & \cdots & a(S|S-1) & a(S|S) \end{pmatrix} \tag{2.1}$$

The matrix is S-by-S where S is the total number of states in the model. The set is called the transition probability matrix.

- **Observation probabilities** (B): B is a matrix which represents output probabilities. The observation probabilities are assumed to be dependent upon state but independent of time t. B(k|i) represents the discrete observation pdf for state i and takes the form k impulses on the real line.

$$B = \begin{pmatrix} b(1|1) & b(1|2) & \cdots & b(1|S-1) & b(1|S) \\ & \ddots & & & \\ & & b(k|i) & & \\ & & & \ddots & \\ b(k|1) & b(k|2) & \cdots & b(k|S-1) & b(k|S) \end{pmatrix} \quad (2.2)$$

S is the total number of states and k is the number of discrete observations in the model.

The set is called the observation probability matrix.

- **Initial distribution** (π(1)): π(1) is the initial probability distribution over states.

  The state probability vector at time t is defined as

$$\pi(t) = \begin{pmatrix} P(x(t)=1) \\ P(x(t)=2) \\ \vdots \\ P(x(t)=S) \end{pmatrix} \quad (2.3)$$

$P(x(t)=i)$ is the probability that the model will be in state i at time t. Some states j may have $P(x(1)=j)=0$, which means that they can not be the initial states.

- **Observation sequence** (O): O represents the possible output observations of the system being modeled. The observation sequence represents the information that is observed from the incoming speech utterance.

Using HMMs for speech recognition includes training the parameters {A, B, π(1)} to match speech observations. The overview of training and speech recognition will be given in the following sections.

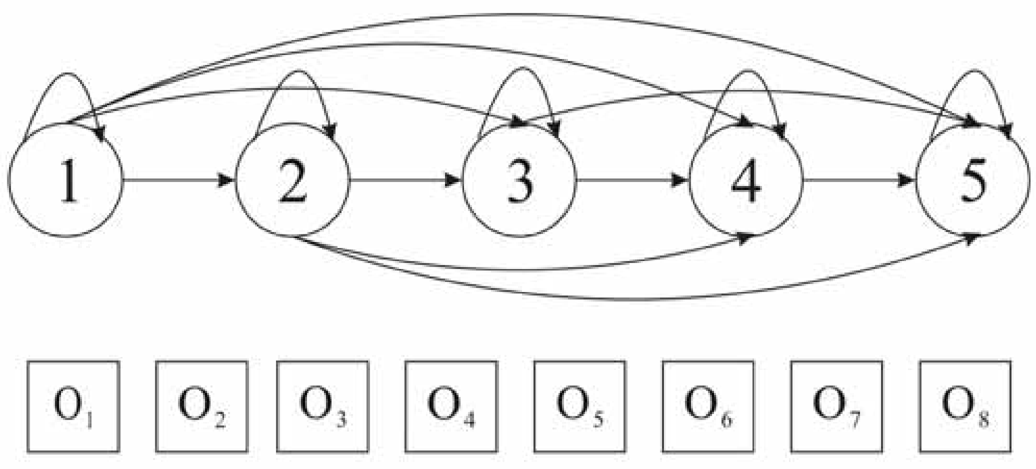

Figure 2.1 Hidden Markov Model for Speech Recognition

2.3 Recognition Using Discrete Observation HMM

There are two key issues associated with the Hidden Markov Model. The first one is the recognition problem. This involves determining the likelihood that each HMM produced an incoming speech observation sequence. The training problem is the second issue. This involves training Hidden Markov Models to represent words by using series of training observation sequence. The recognition problem will be discussed first.

Two measures of likelihood are used in recognition problems, which are "any path" method and "best path" method. We must consider them individually since each of them leads to its own recognition algorithm.

2.3.1 "Any Path" Method

The first method is the "any path" method. It is called "any path" is because the likelihood computed here is based on the probability that the observations could have been produced using any state sequence through the model. One measure of likelihood of a given model m would be $P(y\,|\,m)$, which can be efficiently computed by the forward-backward (F-B) algorithm [15].

At the beginning, we need to define a "forward-going" and a "backward-going" probability sequence [2]. $\alpha(y_1^t, i)$ is defined as the joint probability of having generated the partial forward sequence y_1^t and having arrived at state i at the t th step, given the model m . On the other hand, $\beta\,(y_{t+1}^T, i)$ indicates the probability of generating the backward partial sequence y_{t+1}^T by using model m, given the state sequence appears in state i at time t .

Figure 2.2 shows that there is more than one state "i" at time t through which we can get to j at time t +1, the probabilities should be summed.

$$\alpha(y_1^{t+1}, j) = \sum_{i=1}^{S} \alpha(y_1^t, i) a(j\,|\,i) b(y(t+1)\,|\,j) \tag{2.4}$$

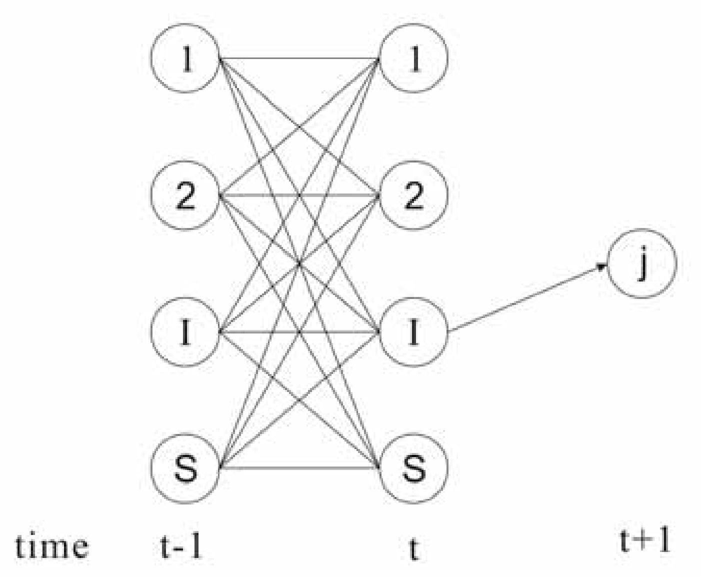

Figure 2.2 Figure of Any Path Method

Finally, the desired likelihood can be obtained at any time in the lattice by summing the F-B products as in equation (2.5).

$$P(y \mid m) = \sum_{i=1}^{S} \alpha(y_1^t, i)\beta(y_{t+1}^T, i) = \sum_{all\_legal\_final\_i} \alpha(y_1^T, i) \tag{2.5}$$

2.3.2 "Best Path" Method

The "Best Path" method is an alternative likelihood measure, which is based on the probability that the Hidden Markov Model could generate the observation sequence using the best possible path. The goal is to find the value $P(y, l^* \mid m)$ where

$$l^* = \arg\max P(y, l \mid m) \tag{2.6}$$

In which l indicates any state sequence of length T. This problem can be considered a sequential optimization problem that is similar to dynamic programming.

The Viterbi algorithm [16] which is used for the "best path" method was introduced by A. J. Viterbi in the context of decoding random sequences. In addition, the algorithm is also called the stochastic form of dynamic programming.

Typically, 10% fewer computations are required with the Viterbi search. However, either the F-B or Viterbi algorithm can generate likelihoods for recognition problems. Both of the methods have been widely used in speech recognition.

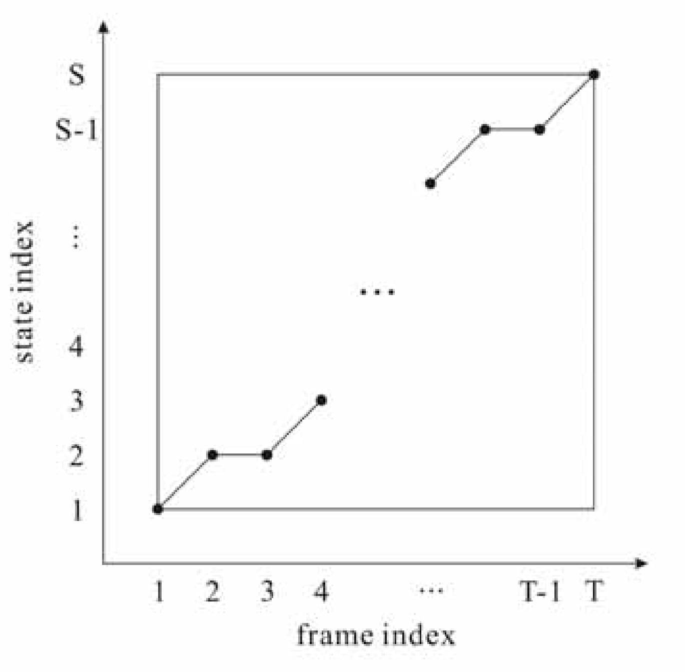

Figure 2.3 HMM Viewed as a Dynamic Programming Problem

2.4 Training Models

The Baum-Welch (F-B) re-estimation algorithm is used for discrete observation Hidden Markov Model training. In addition, the algorithm was developed in a series of papers by Baum and colleagues in the 1960s [8]. It is also called the F-B algorithm because it is based on the forward and backward method which was reviewed in the previous section. Moreover, the goal of training a HMM is to correctly represent its desired word or utterance.

2.4.1 Introduction to Discrete Observation HMM Training

It is assumed that we have a string of the form $y = y_1^T = \{y(1),..., y(T)\}$ taken from a training word, the issue is to use this string to find an appropriate model of form $m = \{S, A, B, \pi(1), O\}$. In the case of training a specific model based on observations from the training data, the F-B algorithm intends to change the parameters $\{A, B, \pi(1)\}$ of the model to find

$$m^* = \arg\max P(y \mid m) \tag{2.6}$$

Initial values for the A and B matrices and for the state probability vector $\pi(1)$ are required at the beginning of the training. There is no known way to exactly compute these quantities from the observation sequence.

After we have the observation sequence and the initial model parameters, we must compute the following four values.

$$\xi(i, j; t) = \begin{cases} \dfrac{\alpha(y_1^t, i) a(j \mid i) b(y(t+1) \mid j) \beta(y_{t+2}^T \mid j)}{P(y \mid m)}, t = 1, \ldots, T-1 \\ 0, otherwise \end{cases} \tag{2.7}$$

$$\gamma(i; t) = \begin{cases} \dfrac{\alpha(y_1^t, i) \beta(y_{t+1}^T \mid i)}{P(y \mid m)}, t = 1, \ldots, T-1 \\ 0, otherwise \end{cases} \tag{2.8}$$

$$v(j; t) = \begin{cases} \dfrac{\alpha(y_1^t, j) \beta(y_{t+1}^T \mid j)}{P(y \mid m)}, t = 1, \ldots, T \\ 0, otherwise \end{cases} \tag{2.9}$$

$$\delta(j, k; t) = \begin{cases} \dfrac{\alpha(y_1^t, j) \beta(y_{t+1}^T \mid j)}{P(y \mid m)}, y(t) = k\_and\_1 \le t \le T \\ 0, otherwise \end{cases} \tag{2.10}$$

Both the forward and backward probability sequences α and β are used extensively in the equations. After that, four related key values need to be computed.

$$\xi(i, j,\bullet) = \sum_{t=1}^{T-1} \xi(i, j; t) \tag{2.11}$$

$$\gamma(i,\bullet) = \sum_{t=1}^{T-1} \gamma(i; t) \tag{2.12}$$

$$v(j,\bullet) = \sum_{t=1}^{T} v(j; t) \tag{2.13}$$

$$\delta(j,k,\bullet) = \sum_{t=1}^{T} \delta(j,k; t) \tag{2.14}$$

Finally, the model's parameters $\{A, B, \pi(1)\}$ can be re-estimated by equations (2.15), (2.16) and (2.17):

$$\bar{a}(j \mid i) = \frac{\xi(i, j;\bullet)}{\gamma(i;\bullet)} \tag{2.15}$$

$$\bar{b}(k \mid j) = \frac{\delta(j,k;\bullet)}{v(j;\bullet)} \tag{2.16}$$

$$P(\underline{x}(1) = i) = \gamma(i;1) \tag{2.17}$$

The algorithm is an iterative computation procedure for estimating a model m, which corresponds to a local maximum of the likelihood $P(y \mid m)$. Also, as the iteration proceeds, the model m has its parameters $\{A, B, \pi(1)\}$ updated to generate a new model m^*. The model will always improve under the re-estimation algorithm unless its parameters have already represented a local maximum.

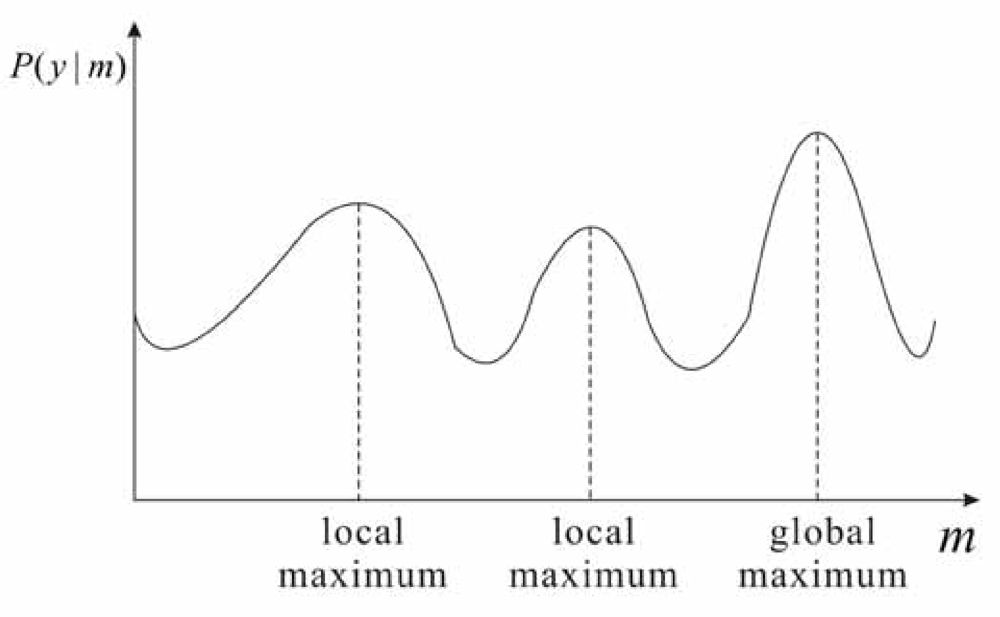

Figure 2.4 The Global Maximum of the HMM Likelihood

$P(y \mid m)$ is generally a nonlinear function, which will definitely have many local maxima and a global maximum in the multidimensional space [17]. However, this procedure does not guarantee to generate the optimal model $m*$. Therefore, it is better to run the algorithm several times with different initial sets of$\{A, B, \pi(1)\}$, and extract the trained model m that yields the largest value of $P(y \mid m)$. The following section will discuss several ways to find the global maximum.

2.4.2 Initial estimates of A and B Matrices

The choice of initial estimates for the elements of the A and B matrices is also a factor of interest for the HMM training. The main problem of training is that although the algorithm is guaranteed to reach a peak of $P(y \mid m)$, the value reached is typically a local maximum. Different starting values of both A and B matrices could yield models with higher or lower values of $P(y \mid m)$. Since the matrices must satisfy these restriction

$$\sum_{j=1}^{N} a_{\text{ij}} = 1 \quad \text{i=1,2,…,N,} \tag{2.18}$$

$$\sum_{k=1}^{M} b_{\text{jk}} = 1 \quad \text{j=1,2,…,M.} \tag{2.19}$$

L.R. Rabiner, S.E. Levinson and M.M. Sondhi brought up an alternative starting condition [5],

$$a_{\text{ij}} = 1/\text{N} + \delta \tag{2.20}$$

$$b_{\text{jk}} = 1/\text{M} + \delta \tag{2.21}$$

where δ is a uniformly distributed random variable whose peak is much smaller than either 1/N or 1/M. A larger local maximum might be obtained by using these initial parameters.

2.4.3 Genetic Algorithms for HMM Training

Many algorithms have been developed to optimize the model parameters to best represent the training observation sequence. However, no single method guarantees to reach the global maximum or other more optimized local maxima.

Since the F-B algorithm starts from an initial guess of parameters, it is better to try different sets of initial $\{A, B, \pi(1)\}$. To improve the training, a stochastic search method called Genetic Algorithm (GA) [6] was introduced by M.Srinivas and Lalit M. Patnaik for HMM training. The Genetic Algorithm imitates natural evolution and performs global searching in the defined searching space. Experiments also showed that GA usually works better than the F-B algorithm.

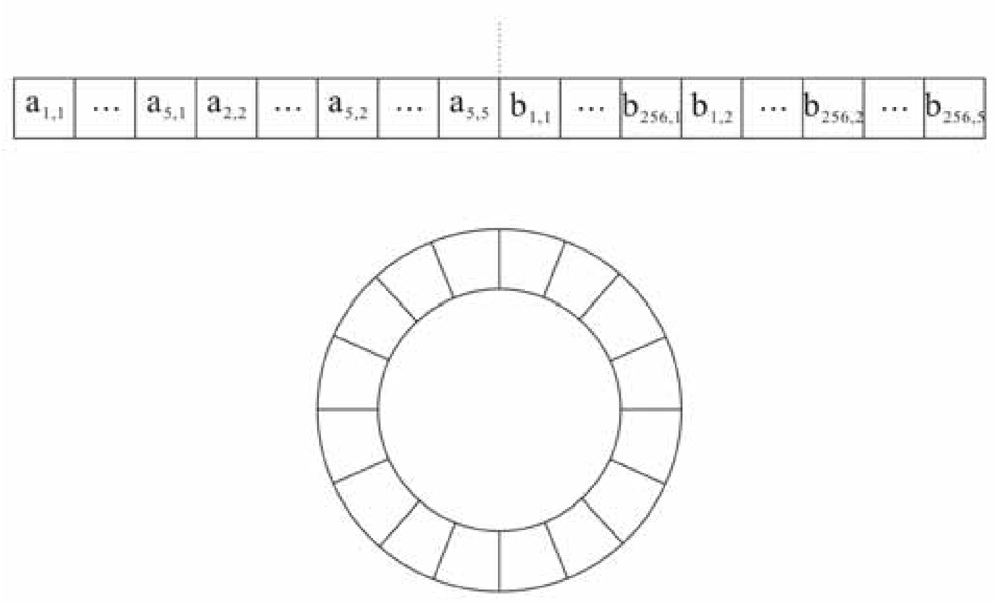

Figure 2.5 Genetic Representation of the HMM Model

Genetic Algorithm Hidden Markov Model training uses the roulette wheel selection scheme as its selection structure [18]. Each solution is distributed to a sector of the roulette wheel with the angle subtended by the sector at the center of the wheel. In other words, the angle is equal to 2π multiplied by the appropriate value of the solution. A solution is selected as an offspring if a random number in the range 0 to 2π falls into the sector corresponds to the solution. The Genetic Algorithm will select solutions by this method until the entire population of the next generation has been produced.

Although experiments have shown that the HMMs trained by the GA can obtain better solutions than by using the F-B algorithm, one of the major drawbacks is that the GA requires lots more computation for global searching before it can converge.

Therefore, in order to improve the efficiency of the Genetic Algorithm, a parallel version of GA called Parallel Genetic Algorithm (PGA) [18] is presented by S. Kwong and C.W. Chau. The results also showed that using PGA for speech recognition provides 18% improvement in recognition rate with the same computation time.

2.4.4 Particle Swarm Optimization for HMM Training

Another method - Particle Swarm Optimization (PSO) [19] has been recently presented for HMM training. The method is designed to estimate optimal parameters of the Hidden Markov Model by finding the global solution or better optimal solutions. As mentioned in the previous section, it is known that the Genetic Algorithm has better results than the F-B algorithm but requires more computation. From the paper "A Particle Swarm Optimization for Hidden Markov model Training" [5], the experiment showed that the PSO-HMM training can provide better results than the GA-HMM training method and the Baum-Welch algorithm. Furthermore, PSO is also more efficient than the Genetic Algorithm since it has a flexible and well-balanced structure to find the global maximum.

The four following equations are used in the PSO-HMM training.

$$f(x_i) = \log P(y \mid m) \tag{2.22}$$

$$pbest_p^k = \arg\max[f(x_p^h)], \quad h = 1, \ldots, k \tag{2.23}$$

$$gbest^k = \arg\max[f(x_p^k)], \quad 1 \le p \le P \tag{2.24}$$

$$\mid f(gbest^k) - f(gbest^{k-1}) \mid < \varepsilon \tag{2.25}$$

The condition of termination is that the maximum number of iteration is reached or the increase of the probability is under the given threshold ε. Finally, *gbest* is assumed to be the HMM parameters $m = \{S, A, B, \pi(1), O\}$ after the optimization process is stopped. Experiments have shown that the average log probabilities of the HMMs trained by PSO have higher values than those trained by Genetic Algorithm and Baum-Welch algorithm.

2.4.5 Training With Multiple Observation Sequences

Usually, training is performed on a large number of separate observation sequences, so it is better to train a Hidden Markov Model with multiple observations [2] in order to provide a more complete representation of the statistical variations likely to be present across utterances. Since we are interested in obtaining speaker-independent models, the observation sequence $y = \{\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \cdots, \mathbf{O}^{(K)}\}$ actually includes several independent sequences $O^{(k)}, k = 1, 2, ..., K$, where $O^{(k)}$ is the training sequence for speaker k [4]. In addition, K is the number of speakers used for training. Moreover, the way to handle multiple observation sequences is to calculate $P(O^k \mid m)$ for each sequence, and maximize the product of the probability using equation (2.26).

$$P = \prod_{k=1}^{K} P(O^{(k)} \mid m) \tag{2.26}$$

The modification of the Baum-Welch algorithm is straightforward for the multiple observation sequences' training. Instead of using equation (2.15) and (2.16), equation (2.27) and (2.28) has been used.

$$\overline{a}(j\,|\,i)=\frac{\sum_{l=1}^{L}\xi^{(l)}(i,j;\bullet)}{\sum_{l=1}^{L}\gamma^{(l)}(i;\bullet)} \tag{2.27}$$

$$\overline{b}(k\,|\,j)=\frac{\sum_{l=1}^{L}\delta^{(l)}(j,k;\bullet)}{\sum_{l=1}^{L}v^{(l)}(j;\bullet)} \tag{2.28}$$

Since the numerator and the denominator of equation (2.27) and (2.28) stand for an average number related to the model, they should be summed by all observations. l represents the result for the lth observation., also, L observation sequences are used in the training. After using the results in equation (2.7)-(2.14), the final equations for adjusting the model parameters are

$$\overline{a}(j\,|\,i)=\frac{\sum_{l=1}^{L}w_k\sum_{t=1}^{T_1-1}\alpha^{(l)}(y_1^t,i)a(j\,|\,i)b(y(t+1)\,|\,j)\beta^{(l)}(y_{t+2}^{T_l}\,|\,j)}{\sum_{l=1}^{L}w_k\sum_{t=1}^{T_1-1}\alpha^{(l)}(y_1^t,i)\beta^{(l)}(y_{t+1}^{T_l}\,|\,i)} \tag{2.29}$$

$$\overline{b}(k\,|\,j)=\frac{\sum_{l=1}^{L}w_k\sum_{\substack{t=1\\ y(t)=k}}^{T_1}\alpha^{(l)}(y_1^t,j)\beta^{(l)}(y_{t+1}^{T_l}\,|\,j)}{\sum_{l=1}^{L}w_k\sum_{t=1}^{T_1-1}\alpha^{(l)}(y_1^t,j)\beta^{(l)}(y_{t+1}^{T_l}\,|\,j)} \tag{2.30}$$

Rabiner proposed using a weight inversely proportional to the probability of the observation sequences, given the model [27].

$$w_k=\frac{1}{P(\mathbf{O}^{(k)}\,|\,m)} \tag{2.31}$$

The weighting method performs well but is not appropriate in all situations since it gives greater weight in the re-estimation to those training data that don't fit the model well. Some other weighting methods are listed below [20] [25].

- Rabiner's vector learning method, $w_k = 1 / P_k$
- Parameter averaging of all models, $w_k = 1 / P_k^{all}$
- Parameter averaging of all models, $w_k = P_k$
- Parameter averaging of all models, $w_k = P_k^{all}$
- Windsorised method
- Direct parameter averaging over the top 50% in terms of $P_{all,k}$, $w_k = P_{all,k}$
- Direct parameter averaging across the best 50% in terms of their $P_{all,k}$ score, $w_k = 1$
- Direct parameter averaging across all models, $w_k = 1$
- Most likely model

No single method is guaranteed to reach the maximum probability. Although we assumed that we can obtain the correct phoneme sequences in the training set, this assumption is not valid in many cases. Therefore, it can be safer to use a weight of $w_k = 1$.

Several papers [21] [22] [23] [24] have presented some more modified methods for HMM with multiple observations based on the Baum-Welch algorithm, which is not the main issue in this section.

2.5 Summary

Recognition and training are the main issues of the Hidden Markov Model. Having completed a background review, this thesis will now focus on multiple observation sequences' training and test how sensitive the model is. The following chapter will be the statement of problem and method of research.

CHAPTER THREE

STATEMENT OF PROBLEM AND METHOD OF RESEARCH

After the background discussion, the main work of the thesis is to simulate a speech recognizer which is trained by different people with different speaking styles. This research also determines how sensitive the training and recognition process is to variations in the training data.

3.1 Introduction of the Model

In this experiment, the configuration of the HMM is a five state left-right model [2]. A set of 256 observation symbols is used, and the length of the observation sequence is 8. In particular, the model is defined by the matrices A and B, and the initial state probability vector, $\boldsymbol{\pi}(1)$, as described below.

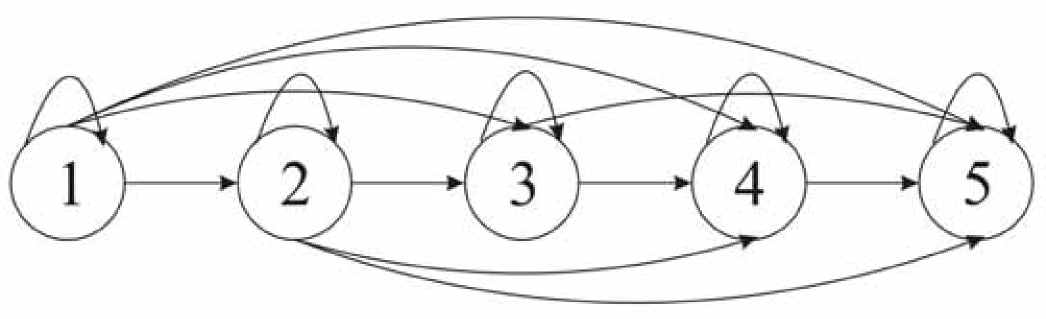

Figure 3.1 A Five States Left-Right Model

Matrix A is a 5-by-5 state transition matrix, its elements at row i, column j are the probabilities A(i|j) of making the transition from state j to state i. Matrix B is a 5-by-256 observation probability matrix, its elements at row k, column j are the probabilities B(j|k) of observation symbol with index j emitted by current state k. The sum of each column in matrices A and B should always be 1. [18]

$\boldsymbol{\pi}(1)$ is the initial state probability vector. The model is assumed to always starts at state 1 and end at state 5, therefore P($\underline{\text{x}}$(1) = 1) = 1, and the probabilities of starting at state 2 to 5 are zero.

Eight sets of observation sequences $y = \{\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \cdots, \mathbf{O}^{(8)}\}$ are trained together during this experiment by the re-estimation formula [20]

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} w_k \sum_{t=1}^{T_k-1} \alpha_t^{(k)}(i) a_{ij} b_j(\mathbf{o}_{t+1}^{(k)}) \beta_{t+1}^{(k)}(j)}{\sum_{k=1}^{K} w_k \sum_{t=1}^{T-1} \alpha_t^{(k)}(i) \beta_t^{(k)}(i)} \tag{3.1}$$

By knowing which w_k fits best for the model, three kinds of w_k are used in this experiment.

- The weight in Rabiner's Vector Learning model method [27]

$$w_k = \frac{1}{P(\mathbf{O}^{(k)} \mid m)} \tag{3.2}$$

- Direct parameter averaging across all models, $w_k = 1$
- Equal weighting based on the mean

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \bar{a}_{ij}^{(k)}}{K} = \frac{\bar{a}_{ij}^{(1)} + \bar{a}_{ij}^{(2)} + \cdots + \bar{a}_{ij}^{(8)}}{8} \tag{3.3}$$

3.2 Initial Estimates of the Parameters

From the overview in chapter 2, it is known that different starting values of both A and B could yield models with higher or lower values of $P(y\,|\,m)$. Since the sum of each column in both of the matrices must be one, the alternative starting values could be

$$a_{\text{ij}} = 1/\text{N} + \delta \text{ x } (1/\text{N}) \tag{3.4}$$

and

$$b_{\text{jk}} = 1/\text{M} + \delta \text{ x } (1/\text{M}), \tag{3.5}$$

where δ is a uniformly distributed random variable which is much smaller than either 1/N or 1/M [4].

After choosing the best w_k for the model, it is also important to estimate the initial parameters. Eight different values of δ will be used to calculate the $P(y\,|\,m)$ in this section.

- The A and B matrices are randomly selected.
- $\delta = 0$
- $\delta = 5\%$
- $\delta = 10\%$
- $\delta = 20\%$
- $\delta = 30\%$
- $\delta = 40\%$
- $\delta = 50\%$

It is necessary to choose the best weighting method and the A and B matrices before re-forming training with multiple observation sequences.

3.3 Multiple Observation Sequences' Training

The 256 possible observation symbols were assumed to be generated by the speech signal after cepstral analysis and vector quantization. In addition, the vector quantization includes 24 elements which are 12 cepstral parameters and 12 delta cepstral parameters.

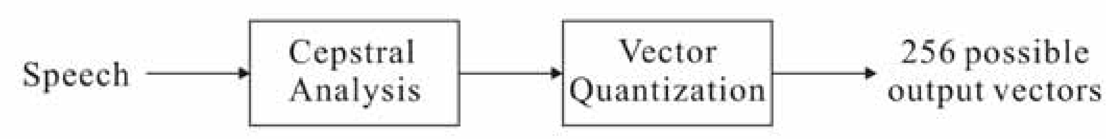

Figure 3.2 System of Generating 256 Possible Output Vectors

It is assumed that if two speech signals have similar cepstra, the signal would have similar representation using vector quantization. Also, it is assumed that the codebook labeling has been performed in such way that two code vectors that are close to each other in vector space have code labels that are numerically close. It can be explained by a 2-dimantional case in figure 3.

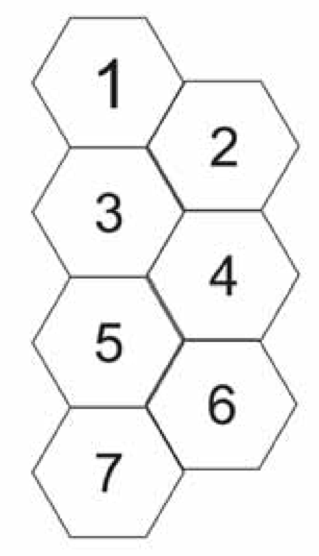

Figure 3.3 Codebook Figure in 2-Dimantional Case

After getting the best parameters as described in section 3.1 and 3.2, we can now train the model with multiple observation sequences. In this section, eight sets of observation sequences close to {16,48,80,112,144,176,208,240} were created to train the model at the same time. Also, the sensitivity of training to variations in the training data is determined by comparing the cases of $\varepsilon = \pm 1$, ± 2 and ± 4. Eight additional sets of observation sequences are generated and their model probabilities by the new A and B matrices were calculated.

Moreover, since {16,48,80,112,144,176,208,240} is assumed to represent the most "standard signal", it is also interesting to vary the observation vectors with the variation restricted to only one of the eight positions in the sequence. This experiment is repeated for each of eight positions in the observation sequence, and check whether the speech signal can be recognized or not.

CHAPTER FOUR

RESULTS

This chapter shows the results of the experiment and provides discussion and conclusions about the data. The first section compares three kinds of weighting methods by showing how well they worked for training using multiple observation sequences. Section 4.2 shows the effect of different initial model parameters. Finally, section 4.3 discusses the main part of the research – how sensitive the Hidden Markov Model is to variations in training data.

4.1 Training with Multiple Observation Sequences

Eight sets of observation sequences $y = \{\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \cdots, \mathbf{O}^{(8)}\}$ would be trained together during this experiment by the re-estimation formula

$$\bar{a}(j \mid i) = \frac{\sum_{l=1}^{L} w_k \sum_{t=1}^{T_1 - 1} \alpha^{(l)}(y_1^t, i) a(j \mid i) b(y(t+1) \mid j) \beta^{(l)}(y_{t+2}^{T_l} \mid j)}{\sum_{l=1}^{L} w_k \sum_{t=1}^{T_1 - 1} \alpha^{(l)}(y_1^t, i) \beta^{(l)}(y_{t+1}^{T_l} \mid i)} \tag{4.1}$$

$$\bar{b}(k \mid j) = \frac{\sum_{l=1}^{L} w_k \sum_{\substack{t=1 \\ y(t)=k}}^{T_1} \alpha^{(l)}(y_1^t, j) \beta^{(l)}(y_{t+1}^{T_l} \mid j)}{\sum_{l=1}^{L} w_k \sum_{t=1}^{T_1 - 1} \alpha^{(l)}(y_1^t, j) \beta^{(l)}(y_{t+1}^{T_l} \mid j)} \tag{4.2}$$

Three kinds of w_k were used in the experiments. Firstly, Rabiner's Vector Learning model method [3] uses the weight

$$w_k = \frac{1}{P(\mathbf{O}^{(k)} \mid m)} \tag{4.3}$$

to train the model by using multiple observation sequences.

Eight sets of observation sequences close to $\{16,48,80,112,144,176,208,240\}$ ($\varepsilon = \pm 1$) were randomly created.

O[1]={16,48,80,112,144,176,208,240}
O[2]={17,47,79,113,145,176,208,241}
O[3]={15,47,79,111,145,177,209,241}
O[4]={16,47,79,112,144,176,209,240}
O[5]={17,49,79,113,144,175,209,241}
O[6]={17,47,79,111,145,176,208,241}
O[7]={16,47,79,112,144,175,207,241}
O[8]={17,49,81,113,143,176,207,239}

Table 4.1 Eight Sets of Observation Sequences

Table 4.2 is the initial model probabilities $P(\mathbf{O}^{(k)} \mid m)$ calculated for each observation sequences where in table 4.1.

P_initial [1]=4.770341788096512E-20
P_initial [2]=6.953773577917202E-20
P_initial [3]=3.256112445226284E-19
P_initial [4]=1.048052547428113E-19
P_initial [5]=3.689518372324197E-20
P_initial [6]=5.548731292852398E-20
P_initial [7]=3.360595980927690E-20
P_initial [8]=8.476924081085776E-21

Table 4.2 The Eight Initial Model Probabilities

P_final[1]~P_final[8], shown below are the model probabilities $P(\mathbf{O}^{(k)} \mid m)$ calculated for each individual observation sequences after training the model by equations (4.1), (4.2) and (4.3). All of the model probabilities increased during the first iteration. However, only one set of model probabilities increased during the second and third iteration while all others decreased. Therefore, the training was unsuccessful even though the average model probability kept increasing.

```
// Iteration #1
                         P_final[1]=2.3700678730789433E-11
                         P_final[2]=7.766249740459384E-9
                         P_final[3]=3.8669030934534955E-13
                         P_final[4]=5.876757236211321E-10
                         P_final[5]=8.199762412730577E-9
                         P_final[6]=1.7436671294990661E-9
                         P_final[7]=1.9720917917193845E-9
                         P_final[8]=1.6631332340896878E-7

                         P_averag=2.332585719700481E-8
        --------------------------------------------------------------------------------
// Iteration #2
                         P_final[1]=1.6886259085191051E-19
                         P_final[2]=3.0242027579993064E-16
                         P_final[3]=3.739251800629411E-5
                         P_final[4]=4.9097614098059785E-14
                         P_final[5]=4.200097166900856E-22
                         P_final[6]=3.4415926183283323E-12
                         P_final[7]=8.097928812990254E-18
                         P_final[8]=2.6345172542854568E-39

                         P_averag=4.674065187161879E-6
        --------------------------------------------------------------------------------

// Iteration #3
                         P_final[1]=1.9497256506268255E-140
                         P_final[2]=1.4311466609236057E-103
                         P_final[3]=2.235081990883815E-196
                         P_final[4]=2.6749554642263157E-137
                         P_final[5]=1.0319774417252415E-90
                         P_final[6]=1.1206234927785088E-130
                         P_final[7]=1.0760985492607772E-134
                         P_final[8]=2.760012067803685E-4

                         P_averag=3.450015084754606E-5
```

Table 4.3 The Model Probabilities After Training by Using eq. (4.1) and (4.2)

The weighting of equation (4.3) gives greater weight in the re-estimation to those utterances that do not fit the model well [28]. Since the observation sequences were generated randomly, the above weighting method was not appropriate. It would be reasonable to use this method if one assumes that the model and data should always have a good fit in training.

Two other methods can be used to train the model. First of all, giving each individual estimate equal weight by computing the mean (equation (4.4)). Secondly, set the weight w_k to 1.

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \bar{a}_{ij}^{(k)}}{K} = \frac{\bar{a}_{ij}^{(1)} + \bar{a}_{ij}^{(2)} + \cdots + \bar{a}_{ij}^{(8)}}{8} \tag{4.4}$$

The same eight sets of observation sequences and initial matrices A and B were used in the training. Results obtained by using this method became reasonable (table 4.4), as their model probabilities increased remarkably and converged after 20 training loops (table 4.4).

```
P_final[1]=2.0599365234375008E-8
P_final[2]=1.5449523925781257E-6
P_final[3]=5.149841308593752E-8
P_final[4]=6.179809570312503E-7
P_final[5]=3.295898437500002E-7
P_final[6]=1.0299682617187503E-6
P_final[7]=4.119873046875002E-7
P_final[8]=4.577636718750002E-9

P_averag=5.013942718505862E-7
```

Table 4.4 Training by Using eq. (4.4)

The model probabilities also increased remarkably and converged after 30 training loops by using the last method. Results were even better than for the previous method since the average model probability became larger (table 4.5).

```
P_final[1]=2.514570951461792E-7
P_final[2]=1.885928213596344E-5
P_final[3]=6.28642737865448E-7
P_final[4]=7.543712854385376E-6
P_final[5]=4.023313522338867E-6
P_final[6]=1.257285475730896E-5
P_final[7]=5.029141902923584E-6
P_final[8]=5.587935447692871E-8

P_averag=6.120535545051098E-6
```

Table 4.5 Training by Using $w_k = 1$

It is usually assumed that utterances involved in the training set are very similar. However, this may not be true in many cases. Also, when dealing with very small values of model probabilities, small changes in $P(\mathbf{O}^{(k)} \mid m)$ can yield large changes in the weights as per equation 4.3. Therefore, it is safer to use a weight of $w_k = 1$.

4.2 Initial Estimates of A and B Matrices

Equation (4.5) and (4.6) represents the initial A and B matrices,

$$a_{\mathrm{ij}} = 1/\mathrm{N} + \delta \mathrm{\ x\ } (1/\mathrm{N}) \tag{4.5}$$

$$b_{\mathrm{jk}} = 1/\mathrm{M} + \delta \mathrm{\ x\ } (1/\mathrm{M}), \tag{4.6}$$

where δ is a uniformly distributed random variable which is much smaller than either 1/N or 1/M [5]. Three randomly selected observation sequences and eight different δ values have been used for training in this section.

The variation in the A and B matrices involved adding $\frac{\delta}{N}$ (or $\frac{\delta}{M}$) and $\frac{-\delta}{N}$ (or $\frac{-\delta}{M}$) in successive terms. Therefore, the A and B matrices are

$$A=\begin{pmatrix} 0.2+\delta/N & 0 & 0 & 0 & 0 \\ 0.2-\delta/N & 0.25+\delta/N & 0 & 0 & 0 \\ 0.2 & 0.25-\delta/N & 0.33+\delta/N & 0 & 0 \\ 0.2+\delta/N & 0.25+\delta/N & 0.33 & 0.5+\delta/N & 0 \\ 0.2-\delta/N & 0.25-\delta/N & 0.33-\delta/N & 0.5-\delta/N & 1 \end{pmatrix} \tag{4.7}$$

and

$$B=\begin{pmatrix} 1/256+\delta/M & \dots & 1/256-\delta/M \\ 1/256-\delta/M & \vdots & 1/256+\delta/M \\ 1/256+\delta/M & \vdots & 1/256-\delta/M \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1/256-\delta/M & \cdots & 1/256+\delta/M \end{pmatrix} \tag{4.8}$$

The experiment shows that the final probability is not significantly dependent on the exact pattern of positive and negative adjustments of A and B elements. δ equals to 0, 5%, 10%, 20%, 30%, 40% and 50% would be used in this experiment.

Observation 1: O = {12,1,3,6,14,2,2,3}

A & B matrices	randomly selected	$\delta = 0$	$\delta = 5\%$	$\delta = 10\%$
Initial P	4.446×10^{-21}	5.241×10^{-20}	4.573×10^{-20}	3.981×10^{-20}
Final P	9.765×10^{-4}	0.015625	0.015625	0.015625

A & B matrices	$\delta = 20\%$	$\delta = 30\%$	$\delta = 40\%$	$\delta = 50\%$
Initial P	2.994×10^{-20}	2.325×10^{-20}	1.761×10^{-20}	1.366×10^{-20}
Final P	0.015625	0.015625	0.00926	0.00926

Observation 2: O = {126,253,38,96,149,234,12,186}

A & B matrices	randomly selected	$\delta = 0$	$\delta = 5\%$	$\delta = 10\%$
Initial P	2.831 x 10^{-19}	5.241 x 10^{-20}	4.314 x 10^{-20}	3.507 x 10^{-20}
Final P	3.2 x 10^{-4}	0.00390625	0.00390625	0.00390625

A & B matrices	$\delta = 20\%$	$\delta = 30\%$	$\delta = 40\%$	$\delta = 50\%$
Initial P	2.205 x 10^{-20}	1.322 x 10^{-20}	7.099 x 10^{-21}	3.424 x 10^{-21}
Final P	0.00390625	0.00390625	0.00390625	0.00390625

Observation 3: O = {12,54,239,97,149,134,3,86}

A & B matrices	randomly selected	$\delta = 0$	$\delta = 5\%$	$\delta = 10\%$
Initial P	5.066 x 10^{-20}	5.241 x 10^{-20}	4.988 x 10^{-20}	4.660 x 10^{-20}
Final P	3.2 x 10^{-4}	0.00390625	0.00390625	0.00390625

A & B matrices	$\delta = 20\%$	$\delta = 30\%$	$\delta = 40\%$	$\delta = 50\%$
Initial P	3.891 x 10^{-20}	2.912 x 10^{-20}	2.068 x 10^{-20}	1.212 x 10^{-20}
Final P	2.441 x 10^{-4}	3.2 x 10^{-4}	3.2 x 10^{-4}	3.2 x 10^{-4}

Table 4.6 Training by Different Sets of Initial A and B Matrices

It can be seen in table 4.6 that the model consistently has the lowest probability after training if the initial A and B matrices are randomly selected. On the other hand, the model has been trained better when δ = 0~10%.

Therefore, selecting δ = 0 would be a good choice in this experiment, and the corresponding A matrix is

$$A = \begin{pmatrix} 0.2 & 0 & 0 & 0 & 0 \\ 0.2 & 0.25 & 0 & 0 & 0 \\ 0.2 & 0.25 & 0.33 & 0 & 0 \\ 0.2 & 0.25 & 0.33 & 0.5 & 0 \\ 0.2 & 0.25 & 0.33 & 0.5 & 1 \end{pmatrix}. \tag{4.9}$$

Also, the B matrix would be uniformly distributed:

$$B = \begin{pmatrix} 1/256 & \ldots & 1/256 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1/256 & \cdots & 1/256 \end{pmatrix}. \tag{4.10}$$

4.3 Test the Effect of Multiple Observation Sequences' Training

After getting the best parameters as described in section 4.1 and 4.2, we can now train the model with multiple observation sequences.

4.3.1 First Experiment

In this section, eight sets of observation sequences close to {16,48,80,112,144,176,208,240} were used to train the model at the same time. Also, the model's sensitivity to variations in training data was evaluated by comparing the cases of ε = ± 1, ± 2 and ± 4. After creating eight new sets of observation sequences and calculating their model probabilities using the new A and B matrices, we can demonstrate how the training works. The weight w_k was set as 1, the A and B matrices are formed as described by equations (4.7) and (4.8) respectively.

Set 1: $\varepsilon = \pm 1$

O[1]={16,48,80,112,144,176,208,240} ("ideal" observation sequence)
O[2]={15,49,80,111,143,177,209,240}
O[3]={15,48,80,112,145,175,209,241}
O[4]={16,48,80,113,145,175,209,239}
O[5]={16,48,81,113,145,177,208,241}
O[6]={16,49,79,113,143,176,208,241}
O[7]={17,47,80,113,143,177,207,241}

```
O[8]={16,48,80,111,143,175,208,239}

P_initial[1]=5.240666512816181E-20
P_initial[2]=5.240666512816181E-20
P_initial[3]=5.240666512816181E-20
P_initial[4]=5.240666512816181E-20
P_initial[5]=5.240666512816181E-20
P_initial[6]=5.240666512816181E-20
P_initial[7]=5.240666512816181E-20
P_initial[8]=5.240666512816181E-20

P_final[1]=1.1175870895385742E-6
P_final[2]=8.046627044677734E-7
P_final[3]=3.0174851417541504E-6
P_final[4]=7.543712854385376E-6
P_final[5]=3.3527612686157227E-6
P_final[6]=1.1920928955078125E-6
P_final[7]=2.682209014892578E-7
P_final[8]=6.705522537231445E-6

P_averag=3.000255674123764E-6
```

Table 4.7. Set 1: $\varepsilon = \pm 1$

O[1]~O[8] are the eight training observation sequences. Since matrices A and B are uniformly distributed, the initial model probabilities would always be the same (5.240666512816181E-20). The average probability increased to $3\text{x}10^{-6}$ after the training. (table 4.7)

After training, eight new observation sequences (O_new[1]~O_new[8]) are created. In addition, P_new[1]~P_new[8] are the new model probabilities by using the trained A and B matrices as shown in the table below.

```
O_new[1]={17,47,80,111,143,175,208,240}
O_new[2]={16,49,80,111,145,175,209,239}
O_new[3]={17,49,80,113,143,175,207,240}
O_new[4]={17,47,81,113,143,175,208,240}
```

```
O_new[5]={16,49,81,112,145,177,207,241}
O_new[6]={15,49,80,113,143,176,208,240}
O_new[7]={16,48,79,113,143,175,208,239}
O_new[8]={16,49,79,112,145,175,208,241}

P_new[1]=2.682209014892578E-7
P_new[2]=1.5087425708770752E-6
P_new[3]=2.682209014892578E-7
P_new[4]=8.940696716308594E-8
P_new[5]=1.6763806343078613E-7
P_new[6]=1.430511474609375E-6
P_new[7]=2.2351741790771484E-6
P_new[8]=6.705522537231445E-7

P_new_averag=8.298084139823914E-7
```

Table 4.8 Set 1: $\varepsilon = \pm 1$, New Average $P(\mathbf{O}^{(k)} \mid m)$ for 8 New Observation Sequences

Although P_new_averag is a little smaller than P_averag, the training can still be considered successful since all probabilities could potentially lead to a recognition decision (table 4.8).

Set 2: $\varepsilon = \pm 2$

```
O[1]={16,48,80,112,144,176,208,240} ("ideal")
O[2]={15,50,78,113,145,178,209,242}
O[3]={16,49,80,114,146,176,207,238}
O[4]={17,49,78,112,142,174,209,241}
O[5]={14,49,82,113,142,177,210,239}
O[6]={16,48,81,112,143,174,209,240}
O[7]={18,50,79,110,142,174,207,240}
O[8]={17,46,79,114,143,178,207,241}

P_final[1]=5.029141902923584E-8
P_final[2]=1.1175870895385742E-8
P_final[3]=5.029141902923584E-8
P_final[4]=4.5262277126312256E-7
P_final[5]=4.190951585769653E-9
```

```
P_final[6]=2.2631138563156128E-7
P_final[7]=7.543712854385372E-8
P_final[8]=4.470348358154296E-8

P_averag=1.1437805369496346E-7

O_new[1]={17,46,81,110,146,175,206,242}
O_new[2]={16,46,78,114,142,177,209,240}
O_new[3]={17,50,82,110,142,176,207,240}
O_new[4]={14,50,79,112,143,176,210,242}
O_new[5]={14,50,82,111,144,178,207,238}
O_new[6]={15,47,81,111,143,175,210,240}
O_new[7]={18,46,81,114,145,174,210,241}
O_new[8]={14,46,79,113,146,175,208,240}

P_new[1]=0.0
P_new[2]=7.543712854385376E-8
P_new[3]=5.029141902923582E-8
P_new[4]=1.1175870895385739E-8
P_new[5]=0.0
P_new[6]=0.0
P_new[7]=2.7939677238464347E-9
P_new[8]=0.0

P_new_averag=1.746229827404022E-8
```

Table 4.9 Set 2: $\varepsilon = \pm 2$, New Average $P(\mathbf{O}^{(k)} \mid m)$ for 8 New Observation Sequences

In the second set, the average of model probabilities is smaller since the observation sequences have larger range. After creating eight new observation sequences to calculate their individual $P(\mathbf{O}^{(k)} \mid m)$ by the trained A and B matrices, there were approximately half of them can be considered as recognizable (The ones with probability = 0 cannot) (table 4.9).

Set 3: $\varepsilon = \pm 4$

```
O[1]={16,48,80,112,144,176,208,240} (“ideal”)
O[2]={17,49,80,111,141,175,208,243}
O[3]={12,49,78,110,141,175,212,243}
O[4]={15,47,83,108,148,174,211,244}
O[5]={17,52,81,113,141,176,206,241}
O[6]={20,48,80,114,141,178,208,244}
O[7]={13,50,77,108,147,179,206,236}
O[8]={14,51,76,113,148,172,212,237}

P_final[1]=8.381903171539307E-9
P_final[2]=1.341104507446289E-7
P_final[3]=1.4901161193847656E-8
P_final[4]=1.862645149230957E-9
P_final[5]=1.4901161193847656E-8
P_final[6]=3.3527612686157227E-8
P_final[7]=9.313225746154785E-10
P_final[8]=1.862645149230957E-9

P_averag=2.6309862732887268E-8

O_new[1]={17,44,78,111,146,176,206,236}
O_new[2]={13,49,84,114,148,174,211,244}
O_new[3]={17,46,82,109,144,172,211,244}
O_new[4]={14,48,78,116,142,175,204,240}
O_new[5]={15,46,79,114,148,175,207,244}
O_new[6]={16,51,77,114,144,179,211,244}
O_new[7]={13,45,78,113,142,177,205,239}
O_new[8]={13,50,77,110,142,175,208,243}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=0.0
P_new[5]=0.0
P_new[6]=4.6566128730773926E-10
P_new[7]=0.0
P_new[8]=0.0

P_new_averag=5.820766091346741E-11
```

Table 4.10 Set 3: $\varepsilon = \pm 4$, New Average $P(\mathbf{O}^{(k)} \mid m)$ for 8 New Observation Sequences

Finally, most of the new observation sequences can not be recognized after the training for the case of $\varepsilon = \pm 4$ since the training sets were not successful because of all the zero-probability cases**.** Consequently, set 1 works the best since the observation sequences fits better in the model (ε is smaller). In addition, two more sets of observations for each ε were used in this experiment ($\varepsilon = \pm 1$ in table 4.11 and 4.12, $\varepsilon = \pm 2$ in table 4.13 and 4.14, $\varepsilon = \pm 4$ in table 4.15 and 4.16). The results are similar to those in table 4.8 - 4.10, and they also restate the conclusion.

The following six tables (4.11-4.16) are the second and third trail for each sets of ε ($\varepsilon = \pm 1$, $\varepsilon = \pm 2$ and $\varepsilon = \pm 4$).

Set 1: $\varepsilon = \pm 1$, second trail

```
O[1]={16,48,80,112,144,176,208,240} ("ideal")
O[2]={16,48,81,113,143,176,208,240}
O[3]={16,49,81,111,145,177,209,240}
O[4]={17,49,81,113,143,177,208,239}
O[5]={15,47,79,111,143,176,208,239}
O[6]={16,49,79,113,144,177,208,239}
O[7]={15,49,80,112,144,177,209,239}
O[8]={17,48,81,112,143,177,209,241}

P_final[1]=2.263113856315613E-6
P_final[2]=6.034970283508301E-6
P_final[3]=1.3411045074462895E-6
P_final[4]=8.940696716308594E-6
P_final[5]=4.470348358154297E-7
P_final[6]=6.705522537231445E-6
P_final[7]=2.0116567611694336E-6
P_final[8]=1.0058283805847168E-6

P_averag=3.5937409847974777E-6

O_new[1]={17,49,80,112,143,176,207,241}
```

```
O_new[2]={15,49,80,111,143,177,209,241}
O_new[3]={15,47,80,112,145,176,209,239}
O_new[4]={15,47,81,112,144,176,209,239}
O_new[5]={15,48,81,113,145,176,209,240}
O_new[6]={15,47,81,111,145,176,208,240}
O_new[7]={15,47,80,112,145,176,209,241}
O_new[8]={15,47,79,113,144,176,207,239}

P_new[1]=0.0
P_new[2]=4.470348358154297E-7
P_new[3]=1.0058283805847172E-7
P_new[4]=6.034970283508301E-7
P_new[5]=4.526227712631227E-7
P_new[6]=1.6763806343078619E-7
P_new[7]=2.514570951461793E-8
P_new[8]=0.0

P_new_averag=2.2456515580415728E-7
```

Table 4.11 Second Trail for $\varepsilon = \pm 1$

Set 1: $\varepsilon = \pm 1$, third trail

```
O[0]={16,48,80,112,144,176,208,240} ("ideal")
O[1]={15,49,79,111,143,175,209,241}
O[2]={16,49,79,112,144,177,209,239}
O[3]={16,47,81,112,145,176,207,240}
O[4]={17,47,79,113,144,177,207,241}
O[5]={16,47,81,112,144,175,207,241}
O[6]={17,47,80,112,144,176,209,241}
O[7]={17,48,81,111,144,176,209,239}

P_final[0]=8.940696716308594E-7
P_final[1]=8.940696716308594E-8
P_final[2]=2.682209014892578E-6
P_final[3]=1.341104507446289E-6
P_final[4]=1.2069940567016597E-6
P_final[5]=8.046627044677734E-6
P_final[6]=1.0728836059570312E-5
P_final[7]=1.6093254089355469E-6

P_averag=3.3248215913772583E-6
```

```
O_new[0]={16,49,79,111,143,177,207,239}
O_new[1]={16,47,79,113,144,177,209,239}
O_new[2]={17,49,81,111,144,175,207,241}
O_new[3]={16,47,80,113,145,177,209,240}
O_new[4]={15,48,81,111,144,176,207,241}
O_new[5]={17,47,81,111,143,176,209,241}
O_new[6]={16,47,79,113,144,175,208,240}
O_new[7]={17,49,80,113,144,177,209,241}

P_new[0]=1.341104507446289E-7
P_new[1]=1.0728836059570308E-6
P_new[2]=1.2069940567016602E-6
P_new[3]=1.1920928955078122E-7
P_new[4]=8.046627044677734E-7
P_new[5]=1.0728836059570312E-6
P_new[6]=2.682209014892577E-7
P_new[7]=5.364418029785154E-7


P_new_averag=6.51925802230835E-7
```

Table 4.12 Third Trail for $\varepsilon = \pm 1$

Set 2: $\varepsilon = \pm 2$, second trail

```
O[1]={16,48,80,112,144,176,208,240} ("ideal")
O[2]={15,48,80,110,142,175,207,239}
O[3]={15,46,81,112,143,177,209,240}
O[4]={14,49,80,112,146,174,208,242}
O[5]={14,49,80,112,144,178,206,238}
O[6]={18,48,79,114,144,178,206,239}
O[7]={15,48,79,111,144,174,210,239}
O[8]={14,47,80,113,142,177,210,241}

P_final[1]=2.980232238769531E-7
P_final[2]=8.381903171539307E-8
P_final[3]=1.1175870895385742E-8
P_final[4]=1.1175870895385742E-7
P_final[5]=4.470348358154297E-7
P_final[6]=8.940696716308594E-8
P_final[7]=2.682209014892578E-7
```

```
P_final[8]=2.7939677238464355E-8

P_averag=1.671724021434784E-7

O_new[1]={17,49,82,111,142,175,206,241}
O_new[2]={15,49,79,112,142,176,207,241}
O_new[3]={18,47,80,113,146,178,210,242}
O_new[4]={14,47,81,113,145,177,210,241}
O_new[5]={15,50,80,114,145,175,208,240}
O_new[6]={15,48,78,111,142,176,209,239}
O_new[7]={16,47,80,111,143,177,208,241}
O_new[8]={15,50,80,111,142,178,206,238}

P_new[1]=0.0
P_new[2]=2.2351741790771484E-8
P_new[3]=4.6566128730773926E-9
P_new[4]=0.0
P_new[5]=0.0
P_new[6]=0.0
P_new[7]=4.6566128730773926E-9
P_new[8]=0.0

P_new_averag=3.958120942115784E-9
```

Table 4.13 Second Trail for $\varepsilon = \pm 2$

Set 2: $\varepsilon = \pm 2$, third trail

```
O[1]={16,48,80,112,144,176,208,240} ("ideal")
O[2]={18,46,80,114,145,178,208,238}
O[3]={14,47,78,110,145,175,209,239}
O[4]={14,47,79,110,143,175,208,242}
O[5]={14,50,79,112,146,177,209,242}
O[6]={16,49,78,114,144,174,207,239}
O[7]={16,48,80,113,142,175,209,238}
O[8]={17,50,81,114,144,178,207,239}

P_final[1]=7.543712854385376E-8
P_final[2]=5.029141902923582E-8
P_final[3]=3.0174851417541504E-7
P_final[4]=1.0058283805847168E-7
P_final[5]=3.3527612686157227E-8
```

```
P_final[6]=7.543712854385376E-8
P_final[7]=7.543712854385376E-8
P_final[8]=5.029141902923584E-8

P_averag=9.534414857625961E-8

O_new[1]={17,48,78,110,146,178,208,238}
O_new[2]={16,49,81,112,142,174,208,240}
O_new[3]={14,49,78,110,144,176,207,242}
O_new[4]={15,48,81,111,144,178,210,241}
O_new[5]={14,50,81,114,144,175,206,238}
O_new[6]={18,49,81,113,145,177,207,242}
O_new[7]={17,46,82,111,145,177,208,240}
O_new[8]={14,49,78,113,145,176,209,240}

P_new[1]=2.2351741790771484E-8
P_new[2]=4.190951585769653E-9
P_new[3]=3.3527612686157227E-8
P_new[4]=0.0
P_new[5]=0.0
P_new[6]=1.862645149230957E-9
P_new[7]=0.0
P_new[8]=8.381903171539307E-9

P_new_averag=8.789356797933578E-9
```

Table 4.14 Third Trail for $\varepsilon = \pm 2$

Set 3: $\varepsilon = \pm 4$, second trail

```
O[1]={16,48,80,112,144,176,208,240} ("ideal")
O[2]={16,51,84,115,143,180,205,241}
O[3]={16,52,78,112,144,178,212,244}
O[4]={19,52,81,115,147,179,207,244}
O[5]={14,46,84,115,142,176,207,243}
O[6]={18,50,84,116,141,173,211,236}
O[7]={12,48,79,115,143,178,209,244}
O[8]={19,49,80,116,142,179,212,238}

P_final[1]=2.2351741790771484E-8
P_final[2]=1.6763806343078613E-8
P_final[3]=6.705522537231445E-8
```

```
P_final[4]=4.470348358154297E-8
P_final[5]=2.2351741790771484E-8
P_final[6]=1.3969838619232178E-9
P_final[7]=2.2351741790771484E-8
P_final[8]=1.4901161193847656E-8

P_averag=2.648448571562767E-8

O_new[1]={14,51,81,115,144,178,205,243}
O_new[2]={15,47,81,108,148,174,209,236}
O_new[3]={16,44,81,116,148,174,208,236}
O_new[4]={19,50,77,116,148,175,205,241}
O_new[5]={17,44,77,113,145,180,209,241}
O_new[6]={18,51,83,113,140,179,206,244}
O_new[7]={19,46,83,109,141,180,211,244}
O_new[8]={13,44,77,111,140,172,208,237}

P_new[1]=3.725290298461914E-9
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=0.0
P_new[5]=0.0
P_new[6]=0.0
P_new[7]=0.0
P_new[8]=0.0


P_new_averag=4.6566128730773926E-10
```

Table 4.15 Second Trail for $\varepsilon = \pm 4$

Set 3: $\varepsilon = \pm 4$, third trail

```
O[1]={16,48,80,112,144,176,208,240} ("ideal")
O[2]={18,44,84,115,143,175,209,236}
O[3]={20,52,81,116,144,177,207,238}
O[4]={13,44,83,113,142,179,211,237}
O[5]={17,46,84,112,142,172,210,237}
O[6]={18,50,81,114,140,176,212,242}
O[7]={14,46,84,115,146,180,204,238}
O[8]={18,44,76,113,146,178,211,242}
```

```
P_final[1]=1.862645149230957E-9
P_final[2]=1.257285475730896E-8
P_final[3]=1.862645149230957E-9
P_final[4]=1.1175870895385742E-8
P_final[5]=1.1175870895385742E-8
P_final[6]=5.587935447692871E-9
P_final[7]=1.1175870895385742E-8
P_final[8]=3.3527612686157227E-8

P_averag=1.1117663234472275E-8

O_new[1]={12,47,79,109,147,180,206,236}
O_new[2]={19,46,81,113,141,173,210,242}
O_new[3]={15,51,79,113,148,178,206,236}
O_new[4]={16,44,83,115,143,180,212,241}
O_new[5]={17,51,77,116,141,174,209,240}
O_new[6]={20,46,81,116,146,174,212,243}
O_new[7]={16,46,78,113,146,177,210,243}
O_new[8]={20,45,81,112,142,180,209,239}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=0.0
P_new[5]=0.0
P_new[6]=0.0
P_new[7]=0.0
P_new[8]=0.0

P_new_averag=0.0
```

Table 4.16 Third Trail for $\varepsilon = \pm 4$

4.3.2 Second Experiment

Since {16,48,80,112,144,176,208,240} is assumed to represent the most "standard signal", it is also interesting to vary the observation vectors with the variation restricted to only one of the eight positions in the sequence. This experiment is repeated for each of eight positions in the observation sequence, and check whether the speech signal can be recognized or not.

Set 1: $\varepsilon = \pm 1$

training set:

```
O[1]={16,48,80,112,144,176,208,240}
O[2]={15,49,80,111,143,177,209,240}
O[3]={15,48,80,112,145,175,209,241}
O[4]={16,48,80,113,145,175,209,239}
O[5]={16,48,81,113,145,177,208,241}
O[6]={16,49,79,113,143,176,208,241}
O[7]={17,47,80,113,143,177,207,241}
O[8]={16,48,80,111,143,175,208,239}

P_initial[1]=5.240666512816181E-20
P_initial[2]=5.240666512816181E-20
P_initial[3]=5.240666512816181E-20
P_initial[4]=5.240666512816181E-20
P_initial[5]=5.240666512816181E-20
P_initial[6]=5.240666512816181E-20
P_initial[7]=5.240666512816181E-20
P_initial[8]=5.240666512816181E-20

P_final[1]=1.1175870895385742E-6
P_final[2]=8.046627044677734E-7
P_final[3]=3.0174851417541504E-6
P_final[4]=7.543712854385376E-6
P_final[5]=3.3527612686157227E-6
P_final[6]=1.1920928955078125E-6
P_final[7]=2.682209014892578E-7
P_final[8]=6.705522537231445E-6

P_averag=3.000255674123764E-6
```

Table 4.17 Training Set for $\varepsilon = \pm 1$

Table 4.17 is the training set for $\varepsilon = \pm 1$, which uses the same training observation sequences as table 4.7. Also, the initial parameters and weighting method are the same as the previous experiment. To reiterate, O[1]~O[8] are the training observation sequences, and P_initial[1]~ P_initial[8] are the $P(\mathbf{O}^{(k)} \mid m)$ calculated by the original A and B matrices. Finally, P_final[1]~ P_final[8] are the final model probabilities calculated by the trained A and B matrices.

case 1:

```
O_new[1]={16,48,80,112,144,176,208,236}
O_new[2]={16,48,80,112,144,176,208,237}
O_new[3]={16,48,80,112,144,176,208,238}
O_new[4]={16,48,80,112,144,176,208,239}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,144,176,208,241}
O_new[7]={16,48,80,112,144,176,208,242}
O_new[8]={16,48,80,112,144,176,208,243}
O_new[9]={16,48,80,112,144,176,208,244}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=1.1175870895385742E-6
P_new[5]=1.1175870895385742E-6
P_new[6]=2.2351741790771484E-6
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0
```

Table 4.18 Case 1 for $\varepsilon = \pm 1$

For example, in table 4.18, O_new[1]~O_new[9] are the new observation sequences that were created to test the model, which are almost the same as {16,48,80,112,144,176,208,240} except a variation of $\varepsilon = \pm 4$ for the last observation component. The above experiment was repeated eight times by changing every column and determining the effect on P_news, which is the $P(\mathbf{O}^{(k)} \mid m)$ by using the new observation sequences with the trained A and B matrices.

case 2:

```
O_new[1]={16,48,80,112,144,176,204,240}
O_new[2]={16,48,80,112,144,176,205,240}
O_new[3]={16,48,80,112,144,176,206,240}
O_new[4]={16,48,80,112,144,176,207,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,144,176,209,240}
O_new[7]={16,48,80,112,144,176,210,240}
O_new[8]={16,48,80,112,144,176,211,240}
O_new[9]={16,48,80,112,144,176,212,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=2.7939677238464355E-7
P_new[5]=1.1175870895385742E-6
P_new[6]=8.381903171539307E-7
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0
```

----

case 3:

```
O_new[1]={16,48,80,112,144,172,208,240}
O_new[2]={16,48,80,112,144,173,208,240}
O_new[3]={16,48,80,112,144,174,208,240}
O_new[4]={16,48,80,112,144,175,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,144,177,208,240}
O_new[7]={16,48,80,112,144,178,208,240}
O_new[8]={16,48,80,112,144,179,208,240}
O_new[9]={16,48,80,112,144,180,208,240}
```

```
P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=1.6763806343078613E-6
P_new[5]=1.1175870895385742E-6
P_new[6]=1.6763806343078613E-6
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0

-------------------------------------------------------
```

case 4:

```
O_new[1]={16,48,80,112,140,176,208,240}
O_new[2]={16,48,80,112,141,176,208,240}
O_new[3]={16,48,80,112,142,176,208,240}
O_new[4]={16,48,80,112,143,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,145,176,208,240}
O_new[7]={16,48,80,112,146,176,208,240}
O_new[8]={16,48,80,112,147,176,208,240}
O_new[9]={16,48,80,112,148,176,208,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=4.470348358154297E-6
P_new[5]=1.1175870895385742E-6
P_new[6]=3.3527612686157227E-6
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0

-------------------------------------------------------
```

case 5:

```
O_new[1]={16,48,80,108,144,176,208,240}
O_new[2]={16,48,80,109,144,176,208,240}
O_new[3]={16,48,80,110,144,176,208,240}
O_new[4]={16,48,80,111,144,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
```

```
O_new[6]={16,48,80,113,144,176,208,240}
O_new[7]={16,48,80,114,144,176,208,240}
O_new[8]={16,48,80,115,144,176,208,240}
O_new[9]={16,48,80,116,144,176,208,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=1.1175870895385742E-6
P_new[5]=1.1175870895385742E-6
P_new[6]=2.2351741790771484E-6
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0

-------------------------------------------------------
```

case 6:

```
O_new[1]={16,48,76,112,144,176,208,240}
O_new[2]={16,48,77,112,144,176,208,240}
O_new[3]={16,48,78,112,144,176,208,240}
O_new[4]={16,48,79,112,144,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,81,112,144,176,208,240}
O_new[7]={16,48,82,112,144,176,208,240}
O_new[8]={16,48,83,112,144,176,208,240}
O_new[9]={16,48,84,112,144,176,208,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=1.862645149230957E-7
P_new[5]=1.1175870895385742E-6
P_new[6]=1.862645149230957E-7
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0

-------------------------------------------------------
```

case 7:

```
O_new[1]={16,44,80,112,144,176,208,240}
```

```
O_new[2]={16,45,80,112,144,176,208,240}
O_new[3]={16,46,80,112,144,176,208,240}
O_new[4]={16,47,80,112,144,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,49,80,112,144,176,208,240}
O_new[7]={16,50,80,112,144,176,208,240}
O_new[8]={16,51,80,112,144,176,208,240}
O_new[9]={16,52,80,112,144,176,208,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=2.2351741790771484E-7
P_new[5]=1.1175870895385742E-6
P_new[6]=4.470348358154297E-7
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0
```

----

case 8:

```
O_new[1]={12,48,80,112,144,176,208,240}
O_new[2]={13,48,80,112,144,176,208,240}
O_new[3]={14,48,80,112,144,176,208,240}
O_new[4]={15,48,80,112,144,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={17,48,80,112,144,176,208,240}
O_new[7]={18,48,80,112,144,176,208,240}
O_new[8]={19,48,80,112,144,176,208,240}
O_new[9]={20,48,80,112,144,176,208,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=0.0
P_new[4]=4.470348358154297E-7
P_new[5]=1.1175870895385742E-6
P_new[6]=2.2351741790771484E-7
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0
```

Table 4.19 Case 2~8 for $\varepsilon = \pm 1$

From the cases in table 4.18 and 4.19, it is interesting to observe that only P_new[4]~ P_new[6] have non-zero values. This is because the training data has $\varepsilon = \pm 1$, and only the observation sequences which has variance less than 1 would have a non-zero value of model probability. For example, there are only 3 different observations in the last column in the training data, which are 239, 240 and 241. Therefore, in case number 1, the model probabilities will have a non-zero value only when the O_news are {16,48,80,112,144,176,208,239}, {16,48,80,112,144,176,208,240} or {16,48,80,112,144,176,208,241}. An additional experiment was performed to support this conclusion.

In this experiment, a single observation sequence was used for training. We are interested in the values of the A and B matrices after training.

O= {12,1,3,6,14,2,2,3}

A matrix:

$$\begin{Bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{Bmatrix}$$

B matrix:

B[12,1] = 1
B[1,2] = 1
B[3,3] = 1
B[6,4]=1
B[2,5]=0.5
B[3,5]=0.25
B[14,5]=0.25
Others Are Zero

Table 4.20 Table Showing the A and B Matrices After Training

It is obvious that a row number which has a non-zero value in the B matrix must have appeared in the observation sequence. For instance, number 12, 1, 3, 6, 2 ,3 and 14 all appear in {12,1,3,6,14,2,2,3}.

Similar results are shown in tables 4.21 and 4.22 for the following observation sequence.

O= {126,253,38,96,149,234,12,186}

A matrix:

$$\left\{\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{matrix}\right\}$$

B matrix:

B[126,1]=1
B[253,2]=1
B[38,3]=1
B[96,4]=1
B[12,5]=0.25
B[149,5]=0.25
B[186,5]=0.25
B[234,5]=0.25
Others Are Zero

Table 4.21 Table Showing the A and B Matrices after Training (2^{nd} Trail)

O= {12,54,239,97,149,134,3,86}

A matrix:

$$\left\{\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{matrix}\right\}$$

B matrix:

B[12,1]=1
B[54,2]=1
B[239,3]=1
B[97,4]=1
B[3,5]=0.25
B[86,5]=0.25
B[134,5]=0.25
B[149,5]=0.25
Others Are Zero

Table 4.22 Table Showing the A and B Matrices after Training (3rd Trail)

Now refer to table 4.18. Since observations 236, 237, 238, 242, 243 and 244 never appeared in the observation sequences, row number 236, 237, 238, 242, 243 and 244 in the B matrix will be zero after the training. Moreover, since

$$P(y\,|\,m) = \sum_{i=1}^{S} \alpha(y_1^t, i)\,\beta(y_{t+1}^T, i) \tag{4.11}$$

and

$$\alpha(y_1^{t+1}, j) = \sum_{i=1}^{S} \alpha(y_1^t, i)a(j\,|\,i)b(y(t+1)\,|\,j), \tag{4.12}$$

the zeros in the B matrix would cause $P(\mathbf{O}^{(k)}\,|\,m)$ to be zero. This is the reason why P_new[4]~ P_new[6] are the only non-zero values in each cases when the training data has variation $\varepsilon = \pm 1$.

After the above explanation, it is time to return to the main experiment for the $\varepsilon = \pm 2$ and $\varepsilon = \pm 4$ case. Table 4.22 is the training set for $\varepsilon = \pm 2$, which uses the same training observation sequences as table 4.9.

Set 2: $\varepsilon = \pm 2$

training set:

```
O[1]={16,48,80,112,144,176,208,240}
O[2]={15,48,80,110,142,175,207,239}
O[3]={15,46,81,112,143,177,209,240}
O[4]={14,49,80,112,146,174,208,242}
O[5]={14,49,80,112,144,178,206,238}
O[6]={18,48,79,114,144,178,206,239}
O[7]={15,48,79,111,144,174,210,239}
O[8]={14,47,80,113,142,177,210,241}

P_initial[1]=5.240666512816181E-20
P_initial[2]=5.240666512816181E-20
P_initial[3]=5.240666512816181E-20
P_initial[4]=5.240666512816181E-20
P_initial[5]=5.240666512816181E-20
P_initial[6]=5.240666512816181E-20
P_initial[7]=5.240666512816181E-20
P_initial[8]=5.240666512816181E-20

P_final[1]=2.980232238769531E-7
P_final[2]=8.381903171539307E-8
P_final[3]=1.1175870895385742E-8
P_final[4]=1.1175870895385742E-7
P_final[5]=4.470348358154297E-7
P_final[6]=8.940696716308594E-8
P_final[7]=2.682209014892578E-7
P_final[8]=2.7939677238464355E-8

P_averag=1.671724021434784E-7
```

Table 4.23 Training Set for $\varepsilon = \pm 2$

case 1:

```
O_new[1]={16,48,80,112,144,176,208,236}
O_new[2]={16,48,80,112,144,176,208,237}
O_new[3]={16,48,80,112,144,176,208,238}
O_new[4]={16,48,80,112,144,176,208,239}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,144,176,208,241}
O_new[7]={16,48,80,112,144,176,208,242}
O_new[8]={16,48,80,112,144,176,208,243}
O_new[9]={16,48,80,112,144,176,208,244}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=1.4901161193847656E-7
P_new[4]=4.470348358154297E-7
P_new[5]=2.980232238769531E-7
P_new[6]=1.4901161193847656E-7
P_new[7]=1.4901161193847656E-7
P_new[8]=0.0
P_new[9]=0.0
```

----

case 2:

```
O_new[1]={16,48,80,112,144,176,204,240}
O_new[2]={16,48,80,112,144,176,205,240}
O_new[3]={16,48,80,112,144,176,206,240}
O_new[4]={16,48,80,112,144,176,207,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,144,176,209,240}
O_new[7]={16,48,80,112,144,176,210,240}
O_new[8]={16,48,80,112,144,176,211,240}
O_new[9]={16,48,80,112,144,176,212,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=2.980232238769531E-7
P_new[4]=1.4901161193847656E-7
P_new[5]=2.980232238769531E-7
P_new[6]=1.4901161193847656E-7
P_new[7]=2.980232238769531E-7
P_new[8]=0.0
P_new[9]=0.0
```

```
          ------------------------------------------------------

case 3:

            O_new[1]={16,48,80,112,144,172,208,240}
            O_new[2]={16,48,80,112,144,173,208,240}
            O_new[3]={16,48,80,112,144,174,208,240}
            O_new[4]={16,48,80,112,144,175,208,240}
            O_new[5]={16,48,80,112,144,176,208,240}
            O_new[6]={16,48,80,112,144,177,208,240}
            O_new[7]={16,48,80,112,144,178,208,240}
            O_new[8]={16,48,80,112,144,179,208,240}
            O_new[9]={16,48,80,112,144,180,208,240}

            P_new[1]=0.0
            P_new[2]=0.0
            P_new[3]=5.960464477539062E-7
            P_new[4]=2.980232238769531E-7
            P_new[5]=2.980232238769531E-7
            P_new[6]=5.960464477539062E-7
            P_new[7]=5.960464477539062E-7
            P_new[8]=0.0
            P_new[9]=0.0

          ------------------------------------------------------

case 4:

            O_new[1]={16,48,80,112,140,176,208,240}
            O_new[2]={16,48,80,112,141,176,208,240}
            O_new[3]={16,48,80,112,142,176,208,240}
            O_new[4]={16,48,80,112,143,176,208,240}
            O_new[5]={16,48,80,112,144,176,208,240}
            O_new[6]={16,48,80,112,145,176,208,240}
            O_new[7]={16,48,80,112,146,176,208,240}
            O_new[8]={16,48,80,112,147,176,208,240}
            O_new[9]={16,48,80,112,148,176,208,240}

            P_new[1]=0.0
            P_new[2]=0.0
            P_new[3]=1.4901161193847656E-7
            P_new[4]=7.450580596923828E-8
            P_new[5]=2.980232238769531E-7
```

```
                P_new[6]=0.0
                P_new[7]=7.450580596923828E-8
                P_new[8]=0.0
                P_new[9]=0.0

             ---------------------------------------------------

case 5:

                O_new[1]={16,48,80,108,144,176,208,240}
                O_new[2]={16,48,80,109,144,176,208,240}
                O_new[3]={16,48,80,110,144,176,208,240}
                O_new[4]={16,48,80,111,144,176,208,240}
                O_new[5]={16,48,80,112,144,176,208,240}
                O_new[6]={16,48,80,113,144,176,208,240}
                O_new[7]={16,48,80,114,144,176,208,240}
                O_new[8]={16,48,80,115,144,176,208,240}
                O_new[9]={16,48,80,116,144,176,208,240}

                P_new[0]=0.0
                P_new[1]=0.0
                P_new[2]=7.450580596923828E-8
                P_new[3]=7.450580596923828E-8
                P_new[5]=2.980232238769531E-7
                P_new[4]=7.450580596923828E-8
                P_new[5]=7.450580596923828E-8
                P_new[6]=0.0
                P_new[7]=0.0

             ---------------------------------------------------
case 6:

                O_new[1]={16,48,76,112,144,176,208,240}
                O_new[2]={16,48,77,112,144,176,208,240}
                O_new[3]={16,48,78,112,144,176,208,240}
                O_new[4]={16,48,79,112,144,176,208,240}
                O_new[5]={16,48,80,112,144,176,208,240}
                O_new[6]={16,48,81,112,144,176,208,240}
                O_new[7]={16,48,82,112,144,176,208,240}
                O_new[8]={16,48,83,112,144,176,208,240}
                O_new[9]={16,48,84,112,144,176,208,240}

                P_new[1]=0.0
                P_new[2]=0.0
```

```
P_new[3]=0.0
P_new[4]=1.1920928955078125E-7
P_new[5]=2.980232238769531E-7
P_new[6]=5.9604644775390625E-8
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0
```

----

case 7:

```
O_new[1]={16,44,80,112,144,176,208,240}
O_new[2]={16,45,80,112,144,176,208,240}
O_new[3]={16,46,80,112,144,176,208,240}
O_new[4]={16,47,80,112,144,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,49,80,112,144,176,208,240}
O_new[7]={16,50,80,112,144,176,208,240}
O_new[8]={16,51,80,112,144,176,208,240}
O_new[9]={16,52,80,112,144,176,208,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=7.450580596923828E-8
P_new[4]=7.450580596923828E-8
P_new[5]=2.980232238769531E-7
P_new[6]=1.4901161193847656E-7
P_new[7]=0.0
P_new[8]=0.0
P_new[9]=0.0
```

----

case 8:

```
O_new[1]={12,48,80,112,144,176,208,240}
O_new[2]={13,48,80,112,144,176,208,240}
O_new[3]={14,48,80,112,144,176,208,240}
O_new[4]={15,48,80,112,144,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={17,48,80,112,144,176,208,240}
```

```
O_new[7]={18,48,80,112,144,176,208,240}
O_new[8]={19,48,80,112,144,176,208,240}
O_new[9]={20,48,80,112,144,176,208,240}

P_new[1]=0.0
P_new[2]=0.0
P_new[3]=8.940696716308594E-7
P_new[4]=8.940696716308594E-7
P_new[5]=2.980232238769531E-7
P_new[6]=0.0
P_new[7]=2.980232238769531E-7
P_new[8]=0.0
P_new[9]=0.0
```

Table 4.24 Case 1~8 for $\varepsilon = \pm 2$

Table 4.23 contains some predictable results. Since the training data has variance $\varepsilon = \pm 2$ and O_new[5] will always represent the "ideal" signal, the $P(\mathbf{O}^{(k)} \mid m)$ have a chance to be a non-zero value between P_new[3] and P_new[7], since $\varepsilon = \pm 2$. However, the probabilities looks smaller than the previous case when $\varepsilon = \pm 1$ because some of the training observations might be a little more different than the ideal one.

The last one would be the $\varepsilon = \pm 4$ case. The training observation sequences would be the same as table 4.10.

Set 3: $\varepsilon = \pm 4$

training set:

```
O[1]={16,48,80,112,144,176,208,240}
O[2]={17,49,80,111,141,175,208,243}
O[3]={12,49,78,110,141,175,212,243}
O[4]={15,47,83,108,148,174,211,244}
O[5]={17,52,81,113,141,176,206,241}
O[6]={20,48,80,114,141,178,208,244}
```

```
O[7]={13,50,77,108,147,179,206,236}
O[8]={14,51,76,113,148,172,212,237}

P_initial[1]=5.240666512816181E-20
P_initial[2]=5.240666512816181E-20
P_initial[3]=5.240666512816181E-20
P_initial[4]=5.240666512816181E-20
P_initial[5]=5.240666512816181E-20
P_initial[6]=5.240666512816181E-20
P_initial[7]=5.240666512816181E-20
P_initial[8]=5.240666512816181E-20

P_final[1]=8.381903171539307E-9
P_final[2]=1.341104507446289E-7
P_final[3]=1.4901161193847656E-8
P_final[4]=1.862645149230957E-9
P_final[5]=1.4901161193847656E-8
P_final[6]=3.3527612686157227E-8
P_final[7]=9.313225746154785E-10
P_final[8]=1.862645149230957E-9

P_averag=2.6309862732887268E-8
```

Table 4.25 Training Set for $\varepsilon = \pm 4$

case 1:

```
O_new[1]={16,48,80,112,144,176,208,236}
O_new[2]={16,48,80,112,144,176,208,237}
O_new[3]={16,48,80,112,144,176,208,238}
O_new[4]={16,48,80,112,144,176,208,239}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,144,176,208,241}
O_new[7]={16,48,80,112,144,176,208,242}
O_new[8]={16,48,80,112,144,176,208,243}
O_new[9]={16,48,80,112,144,176,208,244}

P_new[1]=8.381903171539307E-9
P_new[2]=8.381903171539307E-9
P_new[3]=0.0
P_new[4]=0.0
P_new[5]=8.381903171539307E-9
P_new[6]=8.381903171539307E-9
```

```
                          P_new[7]=0.0
                          P_new[8]=1.6763806343078613E-8
                          P_new[9]=1.6763806343078613E-8

                       -------------------------------------------------------

case 2:

                          O_new[1]={16,48,80,112,144,176,204,240}
                          O_new[2]={16,48,80,112,144,176,205,240}
                          O_new[3]={16,48,80,112,144,176,206,240}
                          O_new[4]={16,48,80,112,144,176,207,240}
                          O_new[5]={16,48,80,112,144,176,208,240}
                          O_new[6]={16,48,80,112,144,176,209,240}
                          O_new[7]={16,48,80,112,144,176,210,240}
                          O_new[8]={16,48,80,112,144,176,211,240}
                          O_new[9]={16,48,80,112,144,176,212,240}

                          P_new[1]=0.0
                          P_new[2]=0.0
                          P_new[3]=5.587935447692871E-9
                          P_new[4]=0.0
                          P_new[5]=8.381903171539307E-9
                          P_new[6]=0.0
                          P_new[7]=0.0
                          P_new[8]=2.7939677238464355E-9
                          P_new[9]=5.587935447692871E-9
                       -------------------------------------------------------

case 3:

                          O_new[1]={16,48,80,112,144,172,208,240}
                          O_new[2]={16,48,80,112,144,173,208,240}
                          O_new[3]={16,48,80,112,144,174,208,240}
                          O_new[4]={16,48,80,112,144,175,208,240}
                          O_new[5]={16,48,80,112,144,176,208,240}
                          O_new[6]={16,48,80,112,144,177,208,240}
                          O_new[7]={16,48,80,112,144,178,208,240}
                          O_new[8]={16,48,80,112,144,179,208,240}
                          O_new[9]={16,48,80,112,144,180,208,240}

                          P_new[1]=4.190951585769653E-9
                          P_new[2]=0.0
                          P_new[3]=4.190951585769653E-9
```

```
P_new[4]=8.381903171539307E-9
P_new[5]=8.381903171539307E-9
P_new[6]=0.0
P_new[7]=4.190951585769653E-9
P_new[8]=4.190951585769653E-9
P_new[9]=0.0

-----------------------------------------------------
```

case 4:

```
O_new[1]={16,48,80,112,140,176,208,240}
O_new[2]={16,48,80,112,141,176,208,240}
O_new[3]={16,48,80,112,142,176,208,240}
O_new[4]={16,48,80,112,143,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,112,145,176,208,240}
O_new[7]={16,48,80,112,146,176,208,240}
O_new[8]={16,48,80,112,147,176,208,240}
O_new[9]={16,48,80,112,148,176,208,240}

P_new[1]=0.0
P_new[2]=3.3527612686157227E-8
P_new[3]=0.0
P_new[4]=0.0
P_new[5]=8.381903171539307E-9
P_new[6]=0.0
P_new[7]=0.0
P_new[8]=8.381903171539307E-9
P_new[9]=1.6763806343078613E-8

-----------------------------------------------------
```

case 5:

```
O_new[1]={16,48,80,108,144,176,208,240}
O_new[2]={16,48,80,109,144,176,208,240}
O_new[3]={16,48,80,110,144,176,208,240}
O_new[4]={16,48,80,111,144,176,208,240}
O_new[5]={16,48,80,112,144,176,208,240}
O_new[6]={16,48,80,113,144,176,208,240}
O_new[7]={16,48,80,114,144,176,208,240}
O_new[8]={16,48,80,115,144,176,208,240}
O_new[9]={16,48,80,116,144,176,208,240}
```

```
            P_new[1]=1.6763806343078613E-8
            P_new[2]=0.0
            P_new[3]=8.381903171539307E-9
            P_new[4]=8.381903171539307E-9
            P_new[5]=8.381903171539307E-9
            P_new[6]=1.6763806343078613E-8
            P_new[7]=8.381903171539307E-9
            P_new[8]=0.0
            P_new[9]=0.0

          ------------------------------------------------------
case 6:

            O_new[1]={16,48,76,112,144,176,208,240}
            O_new[2]={16,48,77,112,144,176,208,240}
            O_new[3]={16,48,78,112,144,176,208,240}
            O_new[4]={16,48,79,112,144,176,208,240}
            O_new[5]={16,48,80,112,144,176,208,240}
            O_new[6]={16,48,81,112,144,176,208,240}
            O_new[7]={16,48,82,112,144,176,208,240}
            O_new[8]={16,48,83,112,144,176,208,240}
            O_new[9]={16,48,84,112,144,176,208,240}

            P_new[1]=2.7939677238464355E-9
            P_new[2]=2.7939677238464355E-9
            P_new[3]=2.7939677238464355E-9
            P_new[4]=0.0
            P_new[5]=8.381903171539307E-9
            P_new[6]=2.7939677238464355E-9
            P_new[7]=0.0
            P_new[8]=2.7939677238464355E-9
            P_new[9]=0.0

          ------------------------------------------------------

case 7:

            O_new[1]={16,44,80,112,144,176,208,240}
            O_new[2]={16,45,80,112,144,176,208,240}
            O_new[3]={16,46,80,112,144,176,208,240}
            O_new[4]={16,47,80,112,144,176,208,240}
            O_new[5]={16,48,80,112,144,176,208,240}
            O_new[6]={16,49,80,112,144,176,208,240}
          O_new[7]={16,50,80,112,144,176,208,240}
          O_new[8]={16,51,80,112,144,176,208,240}
          O_new[9]={16,52,80,112,144,176,208,240}

          P_new[1]=0.0
          P_new[2]=0.0
          P_new[3]=0.0
          P_new[4]=4.190951585769653E-9
          P_new[5]=8.381903171539307E-9
          P_new[6]=8.381903171539307E-9
          P_new[7]=4.190951585769653E-9
          P_new[8]=4.190951585769653E-9
          P_new[9]=4.190951585769653E-9

        ---------------------------------------------------------

case 8:

          O_new[1]={12,48,80,112,144,176,208,240}
          O_new[2]={13,48,80,112,144,176,208,240}
          O_new[3]={14,48,80,112,144,176,208,240}
          O_new[4]={15,48,80,112,144,176,208,240}
          O_new[5]={16,48,80,112,144,176,208,240}
          O_new[6]={17,48,80,112,144,176,208,240}
          O_new[7]={18,48,80,112,144,176,208,240}
          O_new[8]={19,48,80,112,144,176,208,240}
          O_new[9]={20,48,80,112,144,176,208,240}

          P_new[1]=8.381903171539307E-9
          P_new[2]=8.381903171539307E-9
          P_new[3]=8.381903171539307E-9
          P_new[4]=8.381903171539307E-9
          P_new[5]=8.381903171539307E-9
          P_new[6]=1.6763806343078613E-8
          P_new[7]=0.0
          P_new[8]=0.0
          P_new[9]=8.381903171539307E-9
```

Table 4.26 Case 1~8 for $\varepsilon = \pm 4$

It is reasonable to have a non-zero value of $P(\mathbf{O}^{(k)} \mid m)$ from P_new[1]~ P_new[9], since $\varepsilon = \pm 4$ in this set of experiment. However, the model probabilities will be the smallest in the 3 situations (i.e., for the cases of $\varepsilon = \pm 1$, $\varepsilon = \pm 2$ and $\varepsilon = \pm 4$).

Finally, another interesting fact has been discovered. The model probabilities $P(\mathbf{O}^{(k)} \mid m)$ are directly proportional to the number of times the corresponding observation appears in the training set. It can be easily viewed from table 4.26, and it is presented in all cases.

$\varepsilon = \pm 1$, case 1:

Sequence #	O_new[1]	O_new[2]	O_new[3]	O_new[4]	O_new[5]	O_new[6]	O_new[7]	O_new[8]	O_new[9]
8th observation	236	237	238	239	240	241	242	243	244
Appear times	0	0	0	**2**	**2**	**4**	0	0	0
$P(\mathbf{O}^{(k)} \mid m)$	0	0	0	1.1176×10^{-6}	1.1176×10^{-6}	2.2352×10^{-6}	0	0	0
Ratio	0	0	0	**2**	**2**	**4**	0	0	0

$\varepsilon = \pm 2$, case 1:

Sequence #	O_new[1]	O_new[2]	O_new[3]	O_new[4]	O_new[5]	O_new[6]	O_new[7]	O_new[8]	O_new[9]
8th observation	236	237	238	239	240	241	242	243	244
Appear times	0	0	**1**	**3**	**2**	**1**	**1**	0	0
$P(\mathbf{O}^{(k)} \mid m)$	0	0	1.4901×10^{-7}	4.4703×10^{-7}	2.9802×10^{-7}	1.4901×10^{-7}	1.4901×10^{-7}	0	0
Ratio	0	0	**1**	**3**	**2**	**1**	**1**	0	0

$\varepsilon = \pm 4$, case 1:

Sequence #	O_new[1]	O_new[2]	O_new[3]	O_new[4]	O_new[5]	O_new[6]	O_new[7]	O_new[8]	O_new[9]
8th observation	236	237	238	239	240	241	242	243	244
Appear times	**1**	**1**	0	0	**1**	**1**	0	**2**	**2**
$P(\mathbf{O}^{(k)} \mid m)$	8.3819×10^{-9}	8.3819×10^{-9}	0	0	8.3819×10^{-9}	8.3819×10^{-9}	0	1.6764×10^{-8}	1.6764×10^{-8}
Ratio	**1**	**1**	0	0	**1**	**1**	0	**2**	**2**

Table 4.27 Relation Between the Model Probabilities and the Training Observation Sequences.

CHAPTER FIVE

SUMMARY AND CONCLUSIONS

This thesis has described the motivation for recent work in the area of Hidden Markov Model speech recognition and has presented methods for improved multiple observation sequences' HMM training. The goal of the work is to simulate a speech recognizer which is trained by different people with different speaking styles. This research has also investigated how sensitive the training and recognition process is to variations in the training data.

5.1 Discussion of Thesis Work

Firstly, from the overview in chapter 2, it is known that different starting values of both A and B could yield models with higher or lower values of $P(y \mid m)$. Therefore, the choice of initial estimates for the elements of the A and B matrices is an important consideration for the HMM training.

From table 4.6, it can be seen that the model has been trained better when the A and B matrices are:

$$\mathrm{A} = \begin{pmatrix} 0.2 & 0 & 0 & 0 & 0 \\ 0.2 & 0.25 & 0 & 0 & 0 \\ 0.2 & 0.25 & 0.33 & 0 & 0 \\ 0.2 & 0.25 & 0.33 & 0.5 & 0 \\ 0.2 & 0.25 & 0.33 & 0.5 & 1 \end{pmatrix}, \tag{5.1}$$

and

$$\mathrm{B} = \begin{pmatrix} 1/256 & \ldots & 1/256 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1/256 & \cdots & 1/256 \end{pmatrix}. \tag{5.2}$$

Secondly, choosing an appropriate weighting method for multiple observation sequences' training is also a key issue. It is usually assumed that utterances involved in the training set are very similar. However, this may not be true in some cases. Also, when dealing with very small values of model probabilities, small changes in $P(\mathbf{O}^{(k)} \mid m)$ can yield large changes in the weights in Rabiner's vector learning model method. Therefore, it is safer to use a weight of $w_k = 1$.

Thirdly, the set in which speakers speak similar to the standard signal ($\varepsilon = \pm 1$) works the best since the observation sequences fits better in the model (ε is smaller). Furthermore, the recognition can be improved if we increase the amount of training data.

Finally, if recognition is performed on an observation sequence close to the ideal one, it is desired that the range in the training data be larger (ε is larger). Since the training data includes speech spoken by people with different accents, the test data can still be recognized even though the observation sequence is a little different than the ideal one. Although the model probabilities would decrease because the training data has larger range, it can also be improved when the amount of training data increases. In addition, when the observation vectors were varied with the variation restricted to only one of the eight positions in the sequence, the results were very similar in all cases.

Consequently, an ideal case for speech recognition is to recognize a signal which is close to the standard signal by a model trained by a large amount of data with larger range.

5.2 Suggested Directions of Research

From the research, it is known that the speech signal has a higher probability to be recognized if it is similar to the ideal signal. Also, it is encouraged to train the model by more training data with different speaking style. The speech signal which wants to be recognized is not controllable, the only element can be changed is the amount of training data. However, the training set with too much data could be a load to the system. The appropriate amount of data in the training set would be another interesting topic.

BIBLIOGRAPHY

[1] E. Patterson, “Audio-Visual Speech Recognition for Difficult Environments”, 2002

[2] J. R. Deller, Jr., John H. L. Hansen and John G. Proakis, “Discrete-Time Processing of Speech Signals”, 1993

[3] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, Proc. IEEE, Vol. 77, pp. 257-286, 1989

[4] L. R. Rabiner, S. E. Levinson and M. M. Sondhi, “On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition”, The Bell System Technical Journal, 1982

[5] L. Xue, J. Yin and Z. J. L. Jiang, “A Particle Swarm Optimization for Hidden Markov Model Training”, Proceedings on ICSLP, 2006

[6] M. Srinivas and L. M. Patnaik, “Genetic Algorithms: A Survey”, Proceedings of the IEEE, 1994

[7] A. A. Markov, “An Example of Statistical Investigation in the Text of ‘Eugene Onyegin’, Illustrating Coupling of Tests in Chains”, in Proceedings of the Academy of Sciences of St. Petersburg, Russia, 1913

[8] L. E. Baum and T. Petrie, “Statistical Inference for Probabilistic Functions of Finite State Markov Chains”, Annals of Mathematical Statistics, Vol. 37, pp.1554-1563, 1966

[9] K. Seymore, A. McCallum and R. Rosenfeld, “Learning Hidden Markov Model Structure for Information Extraction”, AAAI 99 Workshop on Machine Learning for Information Extraction, (1999)

[10] J. K. Baker, “The DRAGON System – An Overview”, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 23, pp.24-29, Feb. 1975

[11] F. Jelinek, “Continuous Speech Recognition by Statistical Methods”, Proceedings of the IEEE, Vol. 64, pp. 532-556, Apr. 1976

[12] J. D. Ferguson, “Hidden markov analysis: An introduction in Hidden Markov Models for Speech”, Institute for Defense Analyses, Princeton, NJ., 1980

[13] S. E. Levinson, “Structural Methods in Automatic Speech Recognition”, Proceedings of the IEEE, Vol. 73, No. 11, Nov. 1985

[14] D. Jurafsky and J. H. Martin, "Speech and Language Processing", 2000

[15] L. E. Baum and J. A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology", Bulletin of the American Mathematical Society, Vol.73, pp.360-363, 1967

[16] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm", IEEE Transactions of Information Theory, Vol. 13, pp. 260-269, Apr. 1967

[17] L. R. Rabiner, S. E. Levinson and M. M. Sondhi, "An Introduction to Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", The Bell System Technical Journal, Sep. 1982

[18] S. Kwong and C. W. Chau, "Analysis of Parallel Genetic Algorithms on HMM Based Speech Recognition System", IEEE Transactions on Consumer Electronics, Vol. 43, No. 4, Nov. 1997

[19] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, Nov. 1995

[20] R. A. Davis, B. C. Lovell and T. Caelli, "Improved Estimation of Hidden Markov Model Parameters from Multiple Observation Sequences", Proceedings of the 16th International Conference on Pattern Recognition, Vol. 2, 2002

[21] P. M. Baggenstoss, "A Modified Baum-Welch Algorithm for Hidden Markov Models with Multiple Observation Spaces", Proceedings on ICASSP, Vol. 2, June 2000

[22] X. Li, M. Parizeau and R. Plamondon, "Training Hidden Markov Models with Multiple Observations – A Combinatorial Method", IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 22, No. 4, Apr. 2000

[23] P. M. Baggenstoss, "A Modified Baum-Welch Algorithm for Hidden Markov Models with Multiple Observation Spaces", IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 4, May 2001

[24] L. M. Arslan and J. H. L. Hansen, "Selective Training for Hidden Markov Models with Applications to Speech Classification", IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 1, Jan. 1999

[25] N. Liu, R. I. A. Davis, B. C. Lovell and P. J. Kootsookos, "Effect of Initial HMM Choices in Multiple Sequence Training for Gesture Recognition", Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), 2004

[26] X. D. Hoang and J. Hu, “An Efficient Hidden Markov Model Training Scheme for Anomaly Intrusion Detection of Server Applications Based on System Calls”, Proceedings of the 12th IEEE International Conference on Networks, Vol. 2, Nov. 2004

[27] L. R. Rabiner and B. H. Juang, “Fundamentals of Speech Recognition”, New Jersy Prentice Hall, 1993

[28] J. Hosom, “Hidden Markov Models for Speech Recognition”, 2006