This is to certify that the

dissertation entitled

EVALUATION AND IMPROVEMENT OF THE HMM
BY STATE-SPACE MODELING

presented by

Yong-Beom Lee

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Eng

John R. Deller Jr.
Major professor

Date 11 Dec. 2000

DATE DUE	DATE DUE	DATE DUE
		MAR 3 0 2002

EVALUATION AND IMPROVEMENT OF THE HMM BY STATE-SPACE MODELING

By

Yong-Beom Lee

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

2000

ABSTRACT

EVALUATION AND IMPROVEMENT OF THE HMM BY STATE-SPACE MODELING

By

Yong-Beom Lee

Analytical modeling of speech production is not an easy task, in part because of the rapidly time-varying nature of speech signals. The hidden Markov model (HMM) is widely used for the stochastic modeling of time-varying signals, and it has been most applied in the area of speech production and recognition.

Most current HMM research has focused on its applications. On the other hand, studies of the theoretical aspects of the HMM are relatively few. This is due to the difficulties of analyzing a model that is inherently probabilistic and recursive in nature. However, if the fundamentals of the HMM are approached from a different direction, it is possible to obtain useful analyses of the HMM which contribute to its use in speech technologies.

The main objective of this dissertation is to revisit and further investigate three fundamental HMM problems related to speech recognition using a novel mathematical formulation. Rather than the conventional representation of the HMM as a scalar recursive algorithm, the HMM will be represented using a vector-matrix formulation.

It will be shown that the HMM can be represented as a state-space model. The conventional Baum-Welch (time-varying) model as well as an "approximate" time-invariant model will be studied in detail in the context of this new formulation. A more thorough theoretical and empirical investigation of this approximate model is presented in this dissertation. In particular, the spoken-digit recognition problem will be the focus of applied studies.

Some useful results and techniques using the time-invariant approximation of the HMM are addressed and analyzed. In addition, new state-search techniques using clustering, and novel set-membership identification techniques are developed as the basis for a novel HMM training approach. The new training results in HMM state assignments corresponding to acoustically meaningful segmentation of the speech, rather than adherence to the conventional maximum likelihood criterion. The results of new search techniques are compared to those of the Viterbi search.

To my wife and daughter

For their love, support, and sacrifice

ACKNOWLEDGMENTS

I would like to extend my deep thanks and gratitude to Professor John R. Deller, Jr. for his guidance and encouragement of my graduate program for quite a long time. His direction was very important in helping me step into the speech processing and digital signal processing field.

Not only the idea for the state space formulation of the HMM, but also major parts of the problems in this thesis were suggested by Professor Deller, who also made uncountable number of suggestions for the research. Moreover, I am very thankful for his meticulous guidance for the writing.

Also, I would like to thank all the members of my thesis committee: Dr. H. Khalil, Dr. C. Weil, Dr. P. Pierre, and Dr. J. Deller, Jr.

Finally, I give heartful thanks to my family and parents for their love, support, patience, and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Speech is the most natural way of transferring information among human beings. Speech recognition by a machine (e.g., a computer) is a way to translate human speech into corresponding text so that a machine can perform productive work automatically according to human speech inputs. It has many applications such as word dictation, voice activated dialing, automated attendant, command control of a machine, and so forth. The eventual goal of speech processing in engineering is to make machines understand human speech as naturally as humans communicate with one another.

While human communication is natural and easy because of the extraordinary capability of the human brain, speech recognition by a machine is not straightforward in spite of the remarkable development of computer technology.

There are some practical reasons why processing of speech signals is challenging. For example, the same phoneme, when spoken by different speakers, will be acoustically different due to variations in vocal-tract anatomy. Also, the same speaker may produce different versions of the same sound under different circumstances, for instance, when s/he is suffering from a cold and when s/he is not [3]. Certain sounds

may be shortened or completely left out when the speaker talks very fast. Differences in dialect, like phoneme deletion or phoneme substitution, also complicate the speech recognition process. Other problems, like speech related noise (e.g., lips smacks and tongue clicks), add to the difficulty of the recognition systems. It is obvious that without some simplifying assumptions the task of modeling speech for recognition would be highly impractical.

The hidden Markov model (HMM) is a popular technique in many contemporary applications in signal processing, communications, and control. In particular, the HMM has been used to successfully and automatically cope with acoustic uncertainty in speech signal applications [4] using the statistical modeling. For example, achieving a flexible model for rapidly time-varying signals using a dynamic programming technique [4] is very difficult. The popularity of the HMM is due to its simplicity, compactness, and easy implementation. Along with the dynamic time warping technique (DTW), the HMM has been applied to speech recognition systems for many years. In particular, this technique can be globally applied to a large and complex speech recognition system [4].

Although the HMM has been widely researched and extensively applied to the speech processing field, it is true that it lacks diverse formulations so that the HMM can be exploited more efficiently depending on specific applications. This is because of its inherent time-varying nature as well as the somewhat complex recursive structure associated with sophisticated model. HMM research has focused primarily on its application rather than its fundamental characteristics.

1.2 History of the Vector-Matrix Formulations of the HMMs

The conventional *Baum-Welch* algorithm, which is also called the *Forward-Backward* (F-B) algorithm [1, 2, 4], is, a concise, and compact representation of the HMM dynamics for explaining quickly time-varying speech signals efficiently. Whether any-path or a bestpath (Viterbi path) [4] is considered for evaluation/training of the HMM, the conventional F-B approach of the HMM is based on a series of scalar recursion [1, 2, 4]. Also, due to the model characteristics associated with stochastic time-varying signals, the F-B HMM inherently does not have such flexibility and applications as a linear time-invariant model which has stationary parameters representing a model.

Work cited in [6, 7, 8, 27, 30, 31, 32] represents several independent approaches that take advantage of vector-matrix formulations of the HMM. Vector-matrix representations of the HMM provide a diverse and unified way to interpret the HMM operation. In recent paper, by Turin and Karan [27, 31], matrix HMM formulations have been exploited to find useful algorithms for speech recognition technology. In addition, Hjalmarsson *et al.* [30] have used a state-space formulation of HMM to find non-recursive formulae for training the HMM. Similarly, in work by Elliott *et al.* [32], a state-space formulation of HMM has been used for estimation and control. Except for the work conducted in the author's laboratory [6, 7, 8], these vector-matrix formulations are all based on the F-B HMM. Vector-matrix formulations of the *time-invariant approximation* (TIA) of the HMM, which is different from the conventional F-B HMM, were first proposed by Snider and Deller [6, 7].

Turin [27] has proposed vector-matrix representations of the HMM to allow parallel computing to achieve some computational savings during training and evaluation. In particular, he uses vector-matrix formulations to obtain a more computationally

efficient algorithm when speech signals satisfy a certain condition.

Karan *et al.* [31] have used a matrix formulation for the algorithm proposed by Streit [29] in computing the eventual *moments*, defined[1] as $M_{j,i}(k,T) = E\{P_j(O_t)^k\} = \sum_{O_T} P_i(O_T)P_j(O_T)^k$, to measure moments of the output sequence probabilities of the HMM $\mathcal{M}_j$ with respect to $\mathcal{M}_i$. Here $\mathcal{M} = \{N, M, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ is a set of parameter matrices defining the characteristics of the HMM with a state-transition matrix $\boldsymbol{A}$, observation probability matrix $\boldsymbol{B}$, as well as the initial state distribution matrix $\boldsymbol{\pi}$ and the N and M related to the sizes of the matrices. The evaluation $M_{j,i}(k,T)$ proposed in [29] uses vector-matrix descriptions to overcome a computational difficulty by simplifying the recursive scalar computations which have a similar formulation to the *a posteriori* probabilities [1, 2] $P(\mathcal{O}, q_t = i \mid \mathcal{M})$ in the F-B HMM. Such computational savings are possible due to the asymptotic analysis of the dynamics of state-space equations.

Elliott *et al.* [32] suggest a unique state-space model for the two stochastic variables, state and observation, leading to a independent identically distributed (i.i.d) observation process through a change of probability measure. By forming such an ideal distribution, it is possible to obtain several key results related to state estimator by applying the Fubini theorem which allows interchange of expectation and summation in the product measure space. This technique has shown the capability of the state-space structure of the HMM in state estimation and control.

Snider and Deller [6, 7, 8] adopt a simplified probability likelihood measure $\prod_{t=1}^{T} P(O_t)$ to allow a more compact and analyzable approach to evaluate the HMM likelihood, $P(\mathcal{O} \mid \mathcal{M})$. Depending on the circumstances, it is possible to control the *compression index*, the ratio of the number of eigenvalues merged to the total number of eigenvalues in all the HMMs, for trade-off between speech recognition rate and speed and memory complexity requirements.

[1]Precise definitions of notations are established in Chapter 2.

1.3 Problems of Existing Vector-Matrix Formulations of the HMM

In spite of useful results inherent in the vector-matrix and state-space approaches to the HMM cited above, open issues remain.

First, in [27], to obtain a computationally economical formulation with a vector-matrix formulations of the F-B HMM, it is assumed that long stretches of the same symbol string occur within a speech utterance. In fact, such a condition on a speech signal is very helpful to have more computational savings in training and testing of the HMM in reality. For instance, for a very limited small scale system which has a small vocabulary as well as a small number of symbols with simple waveform structures, Turin's condition on signals is justified; therefore, further computational savings can be attained without compromising recognition performance. In addition, under the very unusual circumstance that the speaker is restricted in the number of sounds s/he can reliably produce, Turin's condition is valid even with a relatively large vocabulary.

However, in reality Turin's condition on signals is not ubiquitous in the quickly time-varying speech signals. In practice, even a word model which may have as many as 128 symbols after being quantized for the purpose of efficient, secure storage and transmission, does not usually adhere to Turin's condition very well. Also, for a large scale system with a large vocabulary and complex speech waveform structures, it is unusual to find that Turin's condition is met. Even for a sentence model or a compound model which is composed of concatenating of phones or word HMMs, it is not easy to argue in support of Turin's condition on speech signals. In other words, there is a limitation in applying Turin's algorithms to the general speech signal applications. When speech signals do satisfy Turin's condition, however, computational savings can be obtained. This will be discussed in detail later in the thesis.

The work of Elliott *et al.* [32] is mainly focused on finding an optimal estimation algorithm from observed signals to reveal the originating signals transmitted in a noisy environment. Further, Elliott's derivations are focused mostly on the estimation of states and unknown parameters without a specific procedure for the likelihood evaluation of the HMM. This is a significant derivations from HMM modeling and use in speech recognition.

Snider and Deller report empirically useful results in terms of performance and computation savings, but offer little discussion of the general viability of the modified likelihood $\prod_{t=1}^{T} P(O_t)$. Such a likelihood measure is not identical to $P(\mathcal{O} \mid \mathcal{M})$ of the F-B HMM because of the potential for including extra likelihoods from illegal state paths [4, 7, 25]. A brief explanation about this issue is discussed in [4].

1.4 Objectives

In this research, the focus is on the use of HMMs to model the acoustic process at the lowest levels of a speech recognizer. First, the conventional HMM with a state-space structure will be reformulated leading to a versatile computational structure with rich interpretability. In particular, a TIA HMM suggested in [6, 7, 8] will be a main focus of this dissertation. The viability of the TIA HMM will be shown through several formal approaches. Such time-invariant formulations and corresponding likelihood measures will be argued to be proper approximations of the conventional F-B HMM.

This thesis is composed of five chapters. The present chapter is a short introduction to, and background of, this research. The second chapter introduces time-varying and time-invariant state-space models of the HMM. Vector and matrix formulation notations are used to describe the three fundamental problems of the HMM. In particular, the "illegal state path problem" inherent in the TIA HMM is briefly discussed. The third chapter deals intensively with the problem of illegal state paths produced

by the TIA HMM. A few evolving techniques that reconcile the conventional F-B HMM to the TIA HMM are discussed. Chapter 4 is focused on the problem of finding an appropriate state sequence in some sense for a given speech signal. New state-search techniques using the maximum likelihood criterion, clustering, and novel set-membership identification techniques are developed for HMM training. The results of these search techniques are compared to those of the conventional Viterbi search. The final chapter, Chapter 5, presents research conclusions and future research directions.

In this research, theoretical results are applied mainly to the isolated digit recognition problem, one of the classical problems of speech recognition. For the experimental studies in this research, input speech, which is uttered by an American male speaker, is sampled at a rate of 10kHz. More details of this speech corpus is described in Chapter 3.

Because of the non-stationary nature of speech utterance, the acoustic feature extraction is performed on sampled data on a frame-by-frame basis. Hamming window analysis is applied to each frame, all of which are 25ms long with a 15ms overlap. Then 10^{th} order mel-frequency cepstral coefficients are computed. This produces a sequence of cepstral speech vectors, or as it is usually called, a speech pattern. These speech patterns are classified based on the seven level clusters so that each speech pattern is represented by 128 symbols. It has been implicitly assumed that the given speech signals are free from all background noise so that we can concentrate exclusively on the main modeling problem.

Chapter 2

Vector-Matrix Formulations of the HMM

In this chapter, we first review briefly the HMM theory to support new derivations based on the conventional mathematical formulations. Three basic problems of the HMM, evaluation, estimation (training), and decoding, will be introduced. Then, these basic HMM problems will be reformulated in vector-matrix notation. The conventional F-B algorithm as well as the Viterbi search algorithm will also be subsumed under this vector-matrix formulation. Third, the TIA HMM proposed by Snider and Deller [6, 7, 8] will be discussed, followed by the transformation of the TIA and F-B HMM formulations. Next, some useful characteristics of the HMM discovered using the vector-matrix formulations will be discussed. They give a framework in which to take advantage of the TIA HMM. Fourth, it will be shown analytically with several approaches that such an approximate approach for the HMM evaluation is proper under some practical conditions. Finally, the "illegal path problem" inherent in such an approximation will be briefly discussed. This apparent defect of the TIA HMM will be treated extensively in Chapter 3.

2.1 HMM Background

The HMM was first applied to speech technology independently by Baker [21] at Carnegie Mellon University and Jelinek at IBM in 1975 [1, 2, 3, 4]. When it was first published, neither was it called the HMM, nor was it developed to model and recognize speech signals [22, 23, 24]. However, because of its excellent performance in (apparently) explaining the properties of highly variable signals[1], it has been broadly used in the area of speech signal processing.

The major capability of the HMM lies in its ability to structure the information content of variable data. It also systematically translates this information into a set of stochastic parameters.

The HMM uses a stochastic approach to explain the characterization of speech variability. It is used to model a doubly stochastic production process with the transition parameters modulated by a Markov chain [1, 2]. Thus, the observed speech sequence[2] is assumed to be the result of the interaction of two stochastic processes.

The Markovian assumption on the transition probabilities of the HMM imposes two major constraints on the possible variations in the speech production system. The first constraint is a state model and the other is the dynamics of state transitions according to the Markovian assumption. These allow a compact description of the time-varying speech signal assumed to represent "acoustic states" of speech production.

The HMM is a versatile model which can be used to represent a word, a subword unit, or, in principle, a complete sentence or paragraph [4]. Figure 2.1 shows a typical six-state HMM with *left-to-right* or *Bakis* state transition constraints [4].

Let us now formalize the dynamics of the HMM. Recall the definition of a homo-

[1]This work, in part, investigates whether the HMM accurately models the physical properties of the speech production system. (See Chapter 4.)

[2]Specified later in this section.

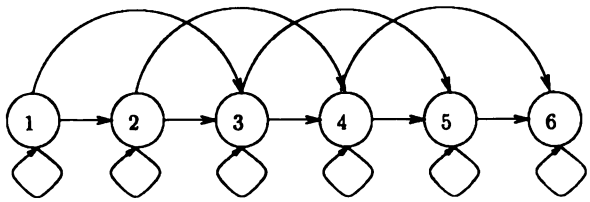

Figure 2.1: Six-state Bakis topology of the HMM.

geneous first-order discrete-state Markov process as one which can be at one of N states[3] $S_1, S_2, \ldots, S_N$ and whose state transitions are dependent only upon the most recent state. Let us denote the state of the system in the abstract at discrete time t by q_t. We denote the stationary conditional probabilities by

$$a_{ji} = P(q_t = S_j | q_{t-1} = S_i), \quad 1 \leq i, j \leq N. \tag{2.1}$$

Additionally, let the initial state probabilities be denoted

$$\pi_i = P(q_1 = S_i), \ 1 \leq i \leq N. \tag{2.2}$$

The realization of the process is a state sequence, say, $\{q_1, q_2, \ldots, q_T\}$. This process is completely characterized by the number of states N, the set of state-transition probabilities $\{a_{ij}\}$, and the set of initial state probabilities $\{\pi_i\}$.

Now consider a discrete-observation *hidden* Markov process. In the discrete HMM, an observation sequence are assumed to be generated by jumping from state to state. With each jump, either during the transitions (on the arc), or upon at the next state, an observation is emitted. At each time, an unknown state emits observation symbol $O_t = k, \ 1 \leq k \leq M$ according to the conditional distribution

$$\begin{aligned} b_j(k) &= P(k \text{ observation at time } t | q_t = S_j), \ 1 \leq j \leq N, \ 1 \leq k \leq M \quad (2.3) \\ &= P(O_t = k \mid q_t = S_j), \end{aligned}$$

[3]By convention, integers are used to represent states as shown in Fig. 2.1.

where M is the number of distinct observation symbols. Symbol O_t is the index of some characteristic measurement extracted from speech, usually frame-wise. Therefore, a speech signal is reduced to strings of features extracted from the acoustic speech waveform. The generated observation sequence, denoted in the abstract by $\mathcal{O} = \{O_1, O_2, \ldots, O_T\}$, is a realization of a *doubly stochastic process*, *i.e.* a random process generated by an unobservable random process. Here, T is the number of observations in the sequence. A model governed by such a probabilistic structure is the *hidden Markov model* when the unobservable random process is a stationary Markov process as described above.

In the remainder of the discussion, we shall use the notation $\mathcal{M}$ to denote the set of elements of an HMM, namely $\mathcal{M} = \{N, M, \{a_{ij}\}, \{b_j(k)\}, \{\pi_i\}\}$.

2.2 Time-Varying Forward-Backward HMM

In this section, we review three HMM problems - evaluation, decoding, and training (or estimation) - and reformulate the conventional F-B HMM in vector-matrix terms. We then exploit this structure to discover interesting properties of the HMM. We also inherently derive some useful expressions for HMM implementation based on state-space formulations.

2.2.1 Evaluation Problem

The recognition or evaluation problem involves the determination of the conditional likelihood for a given observation string, $\mathcal{O}$, namely $P(\mathcal{O} \mid \mathcal{M})$. The most natural measure of the likelihood of a given HMM, say $\mathcal{M}$, in light of observation sequence $\mathcal{O}$, would be *a posteriori* probability $P(\mathcal{M} \mid \mathcal{O})$. However, the available data will not allow us to characterize $P(\mathcal{M} \mid \mathcal{O})$ during the training process, under the condition of equal *a priori* probability $P(\mathcal{M})$ among the HMMs, so it is conventional to take the *a*

posteriori probability $P(\mathcal{O} \mid \mathcal{M})$ as the observation probability measure instead [4].

Vector-Matrix Formulation of the HMM Along a Forward Path

In the F-B solution to this problem [23], the *forward probabilities* are defined by

$$\alpha_i(t) \;=\; P(O_1, O_2, \ldots, O_t, q_t = i \mid \mathcal{M}) \;\text{ for } i = 1, \ldots, N. \tag{2.4}$$

This quantity is the joint probability of the partial observation sequence to time t and residence in state i at time t, given the model $\mathcal{M}$. These probabilities can be calculated recursively as follows [4]: For each state $j = 1, \ldots, N$; and for each $t \geq 1$

$$\alpha_j(t+1) \;=\; \sum_{i=1}^{N} \alpha_i(t) a_{ji} b_j(O_{t+1}) \;\; i = 1, \ldots, N, \tag{2.5}$$

where a_{ji} and $b_j(O_{t+1})$ are defined in (2.1) and (2.3). For the initial time, $\alpha_j(1)$ is defined as $b_j(O_1)\pi_j$. The final conditional observation probability is

$$P(O_1, O_2, \ldots, O_T \mid \mathcal{M}) = \sum_{i=1}^{N} \alpha_i(T). \tag{2.6}$$

The HMM has been developed and studied principally through such conventional formulations. Here, *conventional formulations* implies that in the evaluation and training (explained later) required in the HMM computations, the algorithm is basically focused on individual computations of each state as (2.5) rather than an integrated way that combines state computations. In general, because it represents a linear, time-varying state-space system, the HMM can be researched principally experimentally. However, by combining state processing, it is possible to acquire several significant insights into the HMM which might be difficult to discover otherwise. Once revealed, these useful characteristics of the HMM can be applied to applications for practical benefits.

Generally, matrices provide convenient tools for systemizing laborious calculations by providing a compact notation for describing complicated interrelationships among system variables. Through vector-matrix notations, it is possible to process all HMM states simultaneously and reveal useful properties of the HMM in the process. Let us reformulate the three HMM problems with vector-matrix notations to provide one important basis of this research.

For an N-state HMM, the N recursions of (2.5) for $\alpha_i(t)$, $i = 1, \ldots, N$, can be written in vector-matrix form as

$$\begin{pmatrix} \alpha_1(t+1) \\ \alpha_2(t+1) \\ \vdots \\ \alpha_N(t+1) \end{pmatrix} = \begin{pmatrix} b_1(O_{t+1}) & 0 & \ldots & 0 \\ 0 & b_2(O_{t+1}) & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & b_N(O_{t+1}) \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \ldots & a_{NN} \end{pmatrix} \cdot \begin{pmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \vdots \\ \alpha_N(t) \end{pmatrix}, \quad (2.7)$$

$t = 0, \ldots, T-1.$

This equation can be viewed as the state equation of an N-state state-space model with state variable[4] $\alpha_i(t)$, $i \in [1, N]$. The state equation can be used for $t = 1$ by

[4]Note that the "states" in this context are to denote mathematical variables with which they are used to represent an alternative time-domain dynamics of a HMM. On the other hand, the meaning of "state" in the context of "state model" explaining a HMM is to imply that within such a state, a signal possesses some measurable and distinctive properties.

adding the input term

$$\begin{pmatrix} b_1(O_1) & 0 & \dots & 0 \\ 0 & b_2(O_1) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_N(O_1) \end{pmatrix} \cdot \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{pmatrix} \cdot \delta(t) \tag{2.8}$$

to the right side of (2.7), in which $\delta(t)$ is the Kronecker sequence [4, 33]. In vector-matrix notation, we can write the complete state equation as

$$\boldsymbol{\alpha}(t+1) \;=\; \boldsymbol{\Delta}(t+1)\boldsymbol{A}\boldsymbol{\alpha}(t) + \boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t), \tag{2.9}$$

where the vector and matrix definitions are obvious by correspondence to (2.7) and (2.8).

The output equation for the state-space model is

$$y(t) \;=\; \boldsymbol{C}'\boldsymbol{\alpha}(t). \tag{2.10}$$

The prime in $\boldsymbol{C}'$ is used to denote the matrix transpose. The only output of significance (for making a final decision) is

$$P(O_1, O_2, \dots, O_T \mid \boldsymbol{\mathcal{M}}) \;=\; y(T) = \boldsymbol{C}'\boldsymbol{\alpha}(T), \tag{2.11}$$

with matrix $\boldsymbol{C}$ defined to be a vector of ones,

$$\boldsymbol{C}' \;=\; \mathbf{1}' = [1, 1, \dots, 1]. \tag{2.12}$$

Analysis of HMMs with Vector-Matrix Formulations Along a Forward Path

Note that since the probability of making a transition to *some* state at each time is always unity, then each column of $\boldsymbol{A}$ must sum to one. Accordingly, $\boldsymbol{A}$ is a column stochastic matrix [38] which, in turn, makes it non-negative definite [44, 45]. An important consequence of this is that the vector $\mathbf{1}$ consisting of all ones is a left eigenvector of $\boldsymbol{A}$ with eigenvalue 1 [39] so that $\mathbf{1}'\boldsymbol{A} = \mathbf{1}'$. Non-negative matrices occur in a variety of applications [45]. Non-negativeness implies useful characteristics that can be used to analyze the dynamics of a model [51]. Furthermore, in the *left-to-right* or *Bakis* HMM, which is often employed in speech recognition, $\boldsymbol{A}$ is a lower-triangular matrix with strong diagonal components. In this case, the eigenvalues of $\boldsymbol{A}$ are the diagonal elements themselves. Moreover, because of its triangular structure, if all the eigenvalues are distinct, then eigenvector matrix of $\boldsymbol{A}$ is also triangular. Finding the eigenstructure of such models is therefore relatively computationally inexpensive. The use of this eigenstructure will be explained later in this chapter.

The vector-matrix representation (2.9) and (2.10) reveals interesting results that are not apparent in the usual F-B recursions. Above all, it is the combination of $\boldsymbol{\Delta}(t)$ and $\boldsymbol{A}$, which comprise the effective state-transition part of state equation, monitors and quantifies state path information. Here $\boldsymbol{A}$ has dimensions $N \times N$, and provides all N^2 state-transition probabilities at a given time. This implicitly includes information about whether a given state transition is possible. The premultiplication of $\boldsymbol{A}$ by $\boldsymbol{\Delta}$ regulates the possible paths through the states in light of the states' abilities to generate certain observations. Non-zero values in the diagonal elements of the $\boldsymbol{\Delta}(t)$ matrix allow state jumps at t depending on the locations of those non-zero values whereas zeros prohibit such transitions. For example, if $\boldsymbol{\Delta}_{i,i}(t)$ is zero, meaning that a symbol at time t is not generated from state i, then the i^{th} row of $\boldsymbol{\Delta} \cdot \boldsymbol{A}$ is also zero. Therefore, jumps from any state to state i at time t are prohibited. Consequently,

$\boldsymbol{\Delta}(t)$ can be regarded as a sort of switching matrix at t which specifies the available transitions in accordance with the topology of $\boldsymbol{A}$. Thus, if there is a legal path starting from an initial state to a final state associated with T-duration speech utterance $\boldsymbol{\mathcal{O}} = \{O_1, O_2, \ldots, O_T\}$, T multiplications of matrix pairs $\boldsymbol{\Delta}(t)\boldsymbol{A}$ with $t = 1, \ldots, T$ produce a non-zero matrix. Such a non-zero matrix leads to non-zero likelihood with a suitable initial state probability and a final observation condition.

Henceforth, the model consisting of (2.9) and (2.10) is called the *time-varying state equation* because the composite state-transition matrix, $\boldsymbol{\Delta} \cdot \boldsymbol{A}$, varies with time. The entries in $\boldsymbol{\Delta}$ effectively control the state path by prohibiting entry into a state at time t that cannot produce symbol O_t.

By recursion, the *a posteriori* probability is written in terms of the matrices defined above as

$$P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}) = \boldsymbol{C}' \boldsymbol{\Delta}(T) \boldsymbol{A} \boldsymbol{\Delta}(T-1) \boldsymbol{A} \cdots \boldsymbol{\Delta}(2) \boldsymbol{A} \boldsymbol{\Delta}(1) \boldsymbol{\pi}. \tag{2.13}$$

Since $P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}})$ is a scalar, it can also be expressed as

$$\begin{aligned} P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}}) &= (\boldsymbol{C}' \boldsymbol{\Delta}(T) \boldsymbol{A} \boldsymbol{\Delta}(T-1) \boldsymbol{A} \cdots \boldsymbol{\Delta}(2) \boldsymbol{A} \boldsymbol{\Delta}(1) \boldsymbol{\pi})' \\ &= \boldsymbol{\pi}' \boldsymbol{\Delta}(1) \boldsymbol{A}' \boldsymbol{\Delta}(2) \boldsymbol{A}' \boldsymbol{\Delta}(T-1) \boldsymbol{A}' \boldsymbol{\Delta}(T) \boldsymbol{C}. \end{aligned} \tag{2.14}$$

In fact, this is the formulation with which Turin started in deriving other matrix-based HMM algorithms [27].

Vector-Matrix Formulation of the HMM Along Backward Path

In general, the matrix representation (2.13) provides a flexible way to compute the observation probability through diverse state-space structures and representations for the model. For example, let us derive a state equation that is different from (2.9)-(2.11). To have a state-space model (2.9)-(2.11), $\boldsymbol{\Delta}(t)\boldsymbol{A}$ was considered as a state

variable for (2.13). Instead, let $\boldsymbol{\beta}(t)$ be an N-vector of state variables for (2.14). Define $\boldsymbol{\beta}(T) = \boldsymbol{C}$. Because of recursive nature of equation (2.14), an alternate state-equation-like formulation follows immediately. Let

$$\boldsymbol{\beta}(t) = \boldsymbol{A}^{'} \boldsymbol{\Delta}(t+1)\boldsymbol{\beta}(t+1) \tag{2.15}$$

and $\boldsymbol{\beta}(T) = \boldsymbol{C}$ for $t = T-1, T-2, \ldots, 1$. Then the *a posteriori* conditional observation probability is given by

$$P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}) = \boldsymbol{\pi}^{'} \boldsymbol{\Delta}(1)\boldsymbol{\beta}(1). \tag{2.16}$$

In the matrix formulation, it is not necessary to know the statistical interpretation of $\boldsymbol{\beta}$, whose elements are equivalent to $\beta_i(t)$ in (2.17), but these quantities are recognizable as the *backward probabilities* in the F-B algorithm where they are defined as

$$\beta_i(t) \;=\; P(O_{t+1}, O_{t+2}, \ldots, O_T \mid q_t = i, \boldsymbol{\mathcal{M}}), \quad i = 1, \ldots, N, \tag{2.17}$$

and computed recursively as

$$\beta_i(t) = \sum_{j=1}^{N} a_{ji} b_j(O_{t+1}) \beta_j(t+1), \tag{2.18}$$

similarly to (2.5).

Not surprisingly, the state-space formulation (2.15) of state recursions, written

explicitly as

$$\begin{pmatrix} \beta_1(t) \\ \beta_2(t) \\ \vdots \\ \beta_N(t) \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{N1} \\ a_{12} & a_{22} & \dots & a_{N2} \\ & \vdots & & \\ a_{1N} & a_{2N} & \dots & a_{NN} \end{pmatrix} \tag{2.19}$$

$$\cdot \begin{pmatrix} b_1(O_{t+1}) & 0 & \dots & 0 \\ 0 & b_2(O_{t+1}) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_N(O_{t+1}) \end{pmatrix} \cdot \begin{pmatrix} \beta_1(t+1) \\ \beta_2(t+1) \\ \vdots \\ \beta_N(t+1) \end{pmatrix},$$

$$t = T-1, \dots, 1,$$

can be decomposed into the F-B backward recursions as in (2.18). The output equation complementing (2.19) is given by

$$y(t) = \boldsymbol{\pi}' \boldsymbol{\Delta}(t) \boldsymbol{\beta}(t) \tag{2.20}$$

with the only output of significance (for making a final decision) being

$$\begin{aligned} y(1) &= P(O_1, O_2, \dots, O_T \mid \boldsymbol{\mathcal{M}}) \\ &= \boldsymbol{\pi}' \boldsymbol{\Delta}(1) \boldsymbol{\beta}(1) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_i(1). \end{aligned} \tag{2.21}$$

Result (2.21) is equivalent to the likelihood computation provided by the backward F-B recursion [1, 2, 4].

Other Vector-Matrix Formulations for the HMM

In addition to these results which are equivalent to the widely-used F-B recursions, the matrix formulation provides a flexible state-equation-like structure that serves as

a basis for many other computational structure. For instance, if $\boldsymbol{A}\boldsymbol{\Delta}(t)$ takes to be the state variable rather than $\boldsymbol{\Delta}(t)\boldsymbol{A}$ in (2.13), then we obtain a new state equation with a state vector $\boldsymbol{X}(t)$ governed by the recursion

$$\boldsymbol{X}(t+1) = \boldsymbol{A}\boldsymbol{\Delta}(t)\boldsymbol{X}(t) + \boldsymbol{\pi}\delta(t), \tag{2.22}$$

$$\tag{2.23}$$

with output equation

$$y(t) = P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) = \boldsymbol{C}'\boldsymbol{\Delta}(t)\boldsymbol{X}(t) \tag{2.24}$$

and final likelihood

$$y(T) = P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}) = \boldsymbol{C}'\boldsymbol{\Delta}(T)\boldsymbol{X}(T). \tag{2.25}$$

In fact,

$$\boldsymbol{X}(t+1) = \boldsymbol{A}\boldsymbol{\alpha}(t) \tag{2.26}$$

where $\boldsymbol{\alpha}(t)$ is defined in (2.4).

The expressions above are useful in different circumstances. Usually $\boldsymbol{A}$ has full rank and, thus, almost always has inverse $\boldsymbol{A}^{-1}$. However, in (2.9), $\boldsymbol{\Delta}$ is often singular because of its sparseness. Therefore, having $\boldsymbol{\Delta}$ premultiplied by $\boldsymbol{A}$ in the state equation allows transformation of the state-space equation. Equation (2.15) provides a representation similar to that of (2.22) in the sense that the $\boldsymbol{A}'$ premultiplies $\boldsymbol{\Delta}$ in the state-transition equation. Later we show some useful expressions and properties arising from this characteristic. More novel properties of HMMs will also result from the formulation in which $\boldsymbol{\Delta}$ is premultiplied by $\boldsymbol{A}$ in the state equation.

Some interpretation for the state vector $\boldsymbol{X}(t)$ is obtained by considering one of its elements. From (2.22),

$$\boldsymbol{X}_j(t+1) = \sum_{i=1}^{N} a_{ji} b_i(O_t) X_j(t). \tag{2.27}$$

In the *Moore* form of the HMM [4], a symbol is to be produced when a new state is entered following the transition. This fact makes it difficult to give an exact probabilistic interpretation of $\boldsymbol{X}_j(t+1)$ as is possible for $\boldsymbol{\alpha}(t)$ in (2.4). By inspection, however, state variable $\boldsymbol{X}_j(t+1)$ is similar to $\boldsymbol{\alpha}(t)$ of (2.5) except that $\boldsymbol{X}_j(t+1)$ amounts to *a posteriori* probability before a symbol is produced but after a state transition has occurred. This is similar to Kalman filter update. From (2.27), an observation symbol is apparently generated *before* a state transition occurs, i.e. with the *exit* of the state at time $t-1$. This does not imply that this formulation is equivalent to the *Mealy form* of the HMM [4], since the present expression is based on a_{ji} and b_j parameters from the Moore form. Therefore, the definition of $\boldsymbol{B}$ is quite different in the Mealy form of the HMM which is based on a model in which a symbol is produced *during* the state transition, not upon arrival at a new state. However, the usefulness of (2.27) will be discussed later.

Thus, the state-space formulation suggested here is a general and flexible representation of the HMM of which the conventional F-B HMM is a special case. The representation embodies various interpretations and computational forms for the HMM. There are potentially many interpretations and formulations for the HMM using the vector-matrix form.

Consider another example formulation. Similarly to $\boldsymbol{X}(t)$, we can drive a state-space formulation for the backward computation as

$$\boldsymbol{Y}(t) \;=\; \boldsymbol{\Delta}(t)\boldsymbol{A}'\boldsymbol{Y}(t+1) + \boldsymbol{\Delta}(T)\boldsymbol{C}\delta(T-t) \tag{2.28}$$

$$y(t) = \boldsymbol{\pi}'\boldsymbol{Y}(t) \text{ for } t = T-1, T-2, \ldots, 1 \tag{2.29}$$

and

$$y(T) = P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}) = \boldsymbol{\pi}'\boldsymbol{Y}(1). \tag{2.30}$$

For the new state variable $\boldsymbol{Y}(t)$, the condition $\boldsymbol{Y}(t) = 0$ is imposed for $t > T$.

In terms of the developments above, the final likelihood can be variously represented. For example,

$$\begin{aligned} P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}) &= \boldsymbol{C}'\boldsymbol{\alpha}(T) \\ &= \boldsymbol{\pi}'\boldsymbol{\Delta}(1)\boldsymbol{\beta}(1) \\ &= \boldsymbol{\alpha}'(t)\boldsymbol{\beta}(t), \quad t \in \{1, T\} \\ &= \boldsymbol{\beta}'(t)\boldsymbol{\alpha}(t), \quad t \in \{1, T\} \\ &= \boldsymbol{X}'(t)\boldsymbol{Y}(t), \quad t \in \{1, T\} \\ &= \boldsymbol{Y}'(t)\boldsymbol{X}(t), \quad t \in \{1, T\}. \end{aligned} \tag{2.31}$$

Also, letting $\mathrm{tr}(\cdot)$ denote the trace of a matrix, we can write

$$\begin{aligned} P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}}) &= \mathrm{tr}(\boldsymbol{\alpha}'(t)\boldsymbol{\beta}(t)) \\ &= \mathrm{tr}(\boldsymbol{\alpha}(t)\boldsymbol{\beta}'(t)) \\ &= \mathrm{tr}(\boldsymbol{\Delta}(t)\boldsymbol{A}\ldots\boldsymbol{\Delta}(2)\boldsymbol{A}\boldsymbol{\Delta}(1)\boldsymbol{\pi}(\boldsymbol{A}'\boldsymbol{\Delta}(t+1)\ldots\boldsymbol{A}'\boldsymbol{\Delta}(T)\boldsymbol{C})') \\ &= \mathrm{tr}(\boldsymbol{\Delta}(t)\boldsymbol{A}\ldots\boldsymbol{\Delta}(2)\boldsymbol{A}\boldsymbol{\Delta}(1)\boldsymbol{\pi}\boldsymbol{C}'\boldsymbol{\Delta}(T)\boldsymbol{A}\ldots\boldsymbol{\Delta}(t+1)\boldsymbol{A}). \end{aligned} \tag{2.32}$$

Here $\boldsymbol{\pi}\boldsymbol{C}'$ forms an $N \times N$ matrix.

Interpretation of the HMM Evaluation using the Vector-Matrix Formulation

As yet another example HMM formulation arising from the vector-matrix framework, consider the Bakis HMM structure in which every path starts from a predetermined initial state (by definition, state 1) and finishes at a final state (state N). From equation (2.13), the likelihood can be represented in the compact form as

$$P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) \;=\; \boldsymbol{C}' \boldsymbol{G}(t) \boldsymbol{\pi}, \tag{2.33}$$

where

$$\boldsymbol{G}(t) \;=\; \boldsymbol{\Delta}(t) \boldsymbol{A} \boldsymbol{\Delta}(T-1) \boldsymbol{A} \cdots \boldsymbol{\Delta}(2) \boldsymbol{A} \boldsymbol{\Delta}(1). \tag{2.34}$$

Thus, $\boldsymbol{G}(t)$ amounts to matrix products among vector-matrix-vector multiplications for $P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})$. We know that a matrix is a set of numbers arranged in a rectangular grid of rows and columns. Likewise, matrix $\boldsymbol{G}(t)$ provides an algebraic interpretation for computing $P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}})$ as follows: Let $\boldsymbol{C}' = (0, 0, \ldots, 0, 1)$ and $\boldsymbol{\pi}' = (1, 0, \ldots, 0)$ for simplicity, and let us suppose that the size of each matrix in (2.34) is $N \times N$, and that $\boldsymbol{G}(t)$ is computed in advance. $\boldsymbol{G}(t)$ multiplies both vectors $\boldsymbol{C}'$ and $\boldsymbol{\pi}$ for $P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}})$. Then, the computation of $P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}})$ in (2.33) is equivalent to choosing the $(N, 1)$ element in $\boldsymbol{G}(T)$ according to the position of non-zero entries in $\boldsymbol{C}'$ and $\boldsymbol{\pi}$. Therefore, entry $g_{ji}(t)$ of $\boldsymbol{G}(t)$ amounts to the the sum of likelihoods of the paths leading from state i at initial time to state j at time t. Thus, the vector-matrix representation simplifies the underlying meaning of the forward or backward computation of the HMM in a way which might not be possible with the conventional F-B HMM algorithm. This example shows that the vector-matrix formulation of the HMM elucidates the likelihood computations in

association with state paths for signals.

2.2.2 Decoding Problem

The HMM was conceived as one for which states would represent distinct acoustic phenomena [2, 4]. The solving the decoding problem also elucidates the structure of the model while providing the statistical characteristics of each state.

The state sequence $Q = \{q_1, q_2, \ldots, q_T\}$ corresponding to a speech symbol string $\mathcal{O} = \{O_1, O_2, \ldots, O_T\}$ in the HMM is "hidden." The hidden part of the HMM, a state sequence, must be found based on some modeled way since no exact solution exists. There are several ways of finding a state sequence for a speech signal. Among them, the *Viterbi search* algorithm [57, 58] is popular and recognized as an efficient way of finding an optimal state sequence. Here we structure the Viterbi algorithm in the matrix notation established above and discuss the significance of resulting formulation.

Let

$$d_i(t+1) = \max_{q_1, q_2, \ldots, q_t} P(O_1, O_2, \ldots, O_{t+1}, q_1, q_2, \ldots, q_t, q_{t+1} = i \mid \mathcal{M}), \quad (2.35)$$

which implies the highest probability of a single path ending at state i, at time $t+1$. In the similar way, let

$$\Psi_i(t+1) = \arg\max_{q_t} P(O_1, O_2, \ldots, O_{t+1}, q_1, q_2, \ldots, q_t, q_{t+1} = i \mid \mathcal{M}). \quad (2.36)$$

$\Psi_i(t+1)$ is the state q_t at time t that leads to $d_i(t+1)$. In these terms, the steps of the Viterbi algorithm are as follows:

- **Initialization**

$$\begin{pmatrix} d_1(1) \\ d_2(1) \\ \vdots \\ d_N(1) \end{pmatrix} = \begin{pmatrix} b_1(O_1) & 0 & \dots & 0 \\ 0 & b_2(O_1) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_N(O_1) \end{pmatrix} \cdot \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{pmatrix}, \quad (2.37)$$

$$\begin{pmatrix} \Psi_1(1) \\ \Psi_2(1) \\ \vdots \\ \Psi_N(1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2.38)$$

- **Recursion**

For $t = 2, \dots T$,

$$\begin{aligned} \begin{pmatrix} d_1(t) \\ d_2(t) \\ \vdots \\ d_N(t) \end{pmatrix} &= \begin{pmatrix} b_1(O_t) & 0 & \dots & 0 \\ 0 & b_2(O_t) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_N(O_t) \end{pmatrix} \\ &\times \begin{pmatrix} \max\{d_1(t-1)a_{11}, \dots, d_N(t-1)a_{1N}\} \\ \max\{d_1(t-1)a_{21}, \dots, d_N(t-1)a_{2N}\} \\ \vdots \\ \max\{d_1(t-1)a_{N1}, \dots, d_N(t-1)a_{NN}\} \end{pmatrix} \quad (2.39) \\ &= \begin{pmatrix} b_1(O_t) & 0 & \dots & 0 \\ 0 & b_2(O_t) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_N(O_t) \end{pmatrix} \end{aligned}$$

$$\times \begin{pmatrix} \max_{1\le j\le N}\{d_j(t-1)a_{1j}\} \\ \max_{1\le j\le N}\{d_j(t-1)a_{2j}\} \\ \vdots \\ \max_{1\le j\le N}\{d_j(t-1)a_{Nj}\} \end{pmatrix}, \tag{2.40}$$

$$\begin{pmatrix} \Psi_1(t) \\ \Psi_2(t) \\ \vdots \\ \Psi_N(t) \end{pmatrix} = \begin{pmatrix} \arg\max_{1\le j\le N}\{d_j(t-1)a_{1j}\} \\ \arg\max_{1\le j\le N}\{d_j(t-1)a_{2j}\} \\ \vdots \\ \arg\max_{1\le j\le N}\{d_j(t-1)a_{Nj}\} \end{pmatrix} \tag{2.41}$$

- **Termination**

$$P^*(O_1, O_2, \ldots, O_T \mid \mathcal{M}) = \max\{d_i(T)\}, \tag{2.42}$$

$$q_T^* = \max\{\Psi_i(T)\} \tag{2.43}$$

For $t = T-1, \ldots 1$,

$$q_t^* = \Psi_{q_{t+1}^*}(t+1). \tag{2.44}$$

Where q_t^* is the optimal state at t. We can represent equations (2.37) through (2.41) in matrix form as follows:

$$\boldsymbol{d}(t) = \boldsymbol{\Delta}(t)\max\{\boldsymbol{A} \circ \boldsymbol{d}(t-1)\} \tag{2.45}$$

$$\boldsymbol{\Psi}(t) = \arg\max_j\{\{a_{ij}\}_{1\le i,j\le N} \circ \{d_j(t-1)\}_{1\le j\le N}\} \tag{2.46}$$

for $t = 2, \ldots T$. Where $\boldsymbol{d}(1) = \boldsymbol{\Delta}(1)\boldsymbol{\pi}$ and $\circ$ represents the Hadamard product [36]

of the $N \times N$ matrix and the $N \times 1$ vector defined such that

$$\begin{pmatrix} c_{11} & \dots & c_{1N} \\ c_{21} & \dots & c_{2N} \\ \vdots & \vdots & \vdots \\ c_{N1} & \dots & c_{NN} \end{pmatrix} \circ \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix} = \begin{pmatrix} c_{11}f_1 & \dots & c_{1N}f_N \\ c_{21}f_1 & \dots & c_{2N}f_N \\ \vdots & \vdots & \vdots \\ c_{N1}f_1 & \dots & c_{NN}f_N \end{pmatrix}. \tag{2.47}$$

The likelihood in (2.42) can be represented as

$$P^*(\mathcal{O} \mid \mathcal{M}) = \|\boldsymbol{d}(T)\|_1. \tag{2.48}$$

The back substitution process to find the optimal state path is the same as equation (2.44).

The matrix formulation of the Viterbi search algorithm provides a compact representation of the search algorithm. Also, such formulation makes it easy to implement the search algorithm. In MATLAB, for example, we can get very compact and simplified code.

2.2.3 Training (Estimation) Problem

The training problem concerns how to estimate the elements of $\mathcal{M}$ so as to best describe $\mathcal{O}$. This problem is often solved in a maximum likelihood (ML) framework. To estimate the model parameters given an observation sequence $\mathcal{O}$, the quantity $P(\mathcal{O}|\mathcal{M})$ is optimized. Using an iterative procedure, the model parameter set $\mathcal{M}$ is reestimated to maximize $P(\mathcal{O}|\mathcal{M})$. There are two widely-used algorithms for this optimization problem, the F-B reestimation algorithm (iterative update and improvement) and the Viterbi training algorithm [4, 57, 58]. The Viterbi algorithm has been shown to converge to a proper characterization of the underlying observations [17, 19], and has been found to yield models with comparable performance to those trained

by F-B reestimation [20]. Further, the Viterbi approach is more computationally efficient than the F-B procedure [4].

A matrix representation for the F-B training algorithms is described in [27]. However, an alternative and more straightforward formulation is presented here.

The γ variable [1, 2], a key component in the HMM training, is first represented in a vector-matrix formulation. $\gamma_{ji}(t)$ is the probability of a path being in state i at time t and making a transition to state j at time $t+1$ given $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{M}}$. Thus, we have

$$\begin{aligned} \gamma_{ji}(t) &= P(q_t = i, q_{t+1} = j, \mid \boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{M}}) && (2.49) \\ &= \frac{\alpha_i(t) a_{ji} b_j(O_t) \beta_j(t+1)}{P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})}, \quad t = 1, \ldots, T-1 && (2.50) \end{aligned}$$

for each j and i, where a, b, α, β are defined as (2.1), (2.3), (2.4), and (2.17). Let us define the matrix

$$\boldsymbol{\Gamma}(t) = \begin{pmatrix} \gamma_{11}(t) & \gamma_{12}(t) & \cdots & \gamma_{1N}(t) \\ \gamma_{21}(t) & \gamma_{22}(t) & \cdots & \gamma_{2N}(t) \\ & \vdots & & \\ \gamma_{N1}(t) & \gamma_{N2}(t) & \cdots & \gamma_{NN}(t) \end{pmatrix}. \tag{2.51}$$

Then by (2.50),

$$\boldsymbol{\Gamma}(t) = \frac{1}{P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})} \begin{pmatrix} \beta_1(t+1) & 0 & \cdots & 0 \\ 0 & \beta_2(t+1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta_N(t+1) \end{pmatrix}$$

$$
\times \begin{pmatrix} b_1(O_t) & 0 & \dots & 0 \\ 0 & b_2(O_t) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_N(O_t) \end{pmatrix} \times \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}
$$

$$
\times \begin{pmatrix} \alpha_1(t) & 0 & \dots & 0 \\ 0 & \alpha_2(t) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \alpha_N(t) \end{pmatrix} \tag{2.52}
$$

$$
= \frac{1}{P(\mathcal{O} \mid \mathcal{M})} \begin{pmatrix} \beta_1(t+1) & 0 & \dots & 0 \\ 0 & \beta_2(t+1) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \beta_N(t+1) \end{pmatrix}
$$

$$
\cdot \boldsymbol{\Delta}(t) \boldsymbol{A} \begin{pmatrix} \alpha_1(t) & 0 & \dots & 0 \\ 0 & \alpha_2(t) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \alpha_N(t) \end{pmatrix}, \quad t = 1, \dots, T-1 \tag{2.53}
$$

$$
= \frac{\operatorname{diag}(\boldsymbol{\beta}(t+1)) \cdot (\boldsymbol{\Delta}(t) \boldsymbol{A}) \cdot \operatorname{diag}(\boldsymbol{\alpha}(t))}{P(\mathcal{O} \mid \mathcal{M})}, \quad t = 1, \dots, T-1, \tag{2.54}
$$

where

$$
\operatorname{diag}(\boldsymbol{p}) \triangleq \begin{pmatrix} p_1 & 0 & \dots & 0 \\ 0 & p_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & p_N \end{pmatrix} \tag{2.55}
$$

for any row or column vector $\boldsymbol{p}$.

Then, if $\gamma_i(t)$ (note single subscript) is defined as in [40],

$$\gamma_i(t) = P(q_t = i \mid \mathcal{O}, \mathcal{M}), \tag{2.56}$$

we have

$$\boldsymbol{\gamma}(t) = \begin{pmatrix} \gamma_1(t) \\ \gamma_2(t) \\ \vdots \\ \gamma_N(t) \end{pmatrix} = \frac{1}{(\boldsymbol{\alpha}'(t)\boldsymbol{\beta}(t))} \begin{pmatrix} \alpha_1(t)\beta_1(t) \\ \alpha_2(t)\beta_2(t) \\ \vdots \\ \alpha_N(t)\beta_N(t) \end{pmatrix}, \quad t = 1, \ldots, T-1 \tag{2.57}$$

$$= \boldsymbol{\Gamma}'(t) \cdot \mathbf{1} \tag{2.58}$$

assuming that $\boldsymbol{\alpha}'(t)\boldsymbol{\beta}(t) \neq 0$. This equation holds for any $t \in [1, T-1]$.

In terms of variables defined above, the reestimation formula for the state-transition matrix $\boldsymbol{A}$ becomes

$$\hat{\boldsymbol{A}} = \sum_{t=1}^{T-1} \boldsymbol{\Gamma}(t) \left(\sum_{t=1}^{T-1} \begin{pmatrix} \gamma_1(t) & 0 & \ldots & 0 \\ 0 & \gamma_2(t) & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \gamma_N(t) \end{pmatrix} \right)^{-1} \tag{2.59}$$

$$= \sum_{t=1}^{T-1} \boldsymbol{\Gamma}(t) \left(\sum_{t=1}^{T-1} \mathrm{diag}(\boldsymbol{\gamma}(t)) \right)^{-1} \tag{2.60}$$

in which $\sum_{t=1}^{T-1}$ denotes a element-by-element conventional matrix summation. Here $\hat{\boldsymbol{A}}$ denotes estimated value of $\boldsymbol{A}$.

To reestimate the $\boldsymbol{B}$ matrix, let $\boldsymbol{V} = \{v_{kj}\}$ be an $K \times N$ matrix such that

$$\boldsymbol{V} = \{ \sum_{t \text{ s.t. } O_t = k} \gamma_j(t) \}_{1 \leq k \leq K,\ 1 \leq j \leq N}. \tag{2.61}$$

Then, with the matrix defined above, the reestimation formula for the observation

matrix $\boldsymbol{B}$ becomes

$$\hat{\boldsymbol{B}} = \boldsymbol{V} \times \left(\sum_{t=1}^{T} \begin{pmatrix} \gamma_1(t) & 0 & \dots & 0 \\ 0 & \gamma_2(t) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \gamma_N(t) \end{pmatrix} \right)^{-1} \tag{2.62}$$

$$= \boldsymbol{V} \times \left(\sum_{t=1}^{T} \mathrm{diag}(\boldsymbol{\gamma}(t)) \right)^{-1}. \tag{2.63}$$

As a consequence, we have a compact expressions of its training algorithm with matrix formulations.

2.3 Time-Invariant Approximation for the HMM

A vector-matrix formulation of the F-B HMM is posed in a state-space formulation with suitable state variables, state-transition matrix, and output equation. In modeling terms, the state-space system of the F-B HMM is linear, but time-varying. However, because the state-transition matrix is time varying due to varying observation symbol probabilities, unlike the linear time-invariant model, it is not easy to transform the F-B HMM expressions and to derive other representations for the HMM which may be useful to find various techniques to solve the HMM problems for some speech applications.

An approximate time-invariant model for the time-varying state-space F-B HMM with more potential for application is presented. Since theoretically it is not possible to make the time-varying F-B HMM and TIA HMM identical, a revised formulation with different state variables and likelihood measurements needs to be posed for the approximation. There are several ways to pose such an approximation. Here, we study one approximation method based on a state-space formulation which was developed

in the author's laboratory by Snider and Deller [4, 6, 7].

The original motivation of the technique proposed in [6, 7] was to decrease the computational load in evaluating the HMMs. However, here we will show that such a derivation is useful not only for the computational aspect, but also for an reasonable approximation of the likelihood $P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})$ computed using the time-invariant state-space model. Practically, in the HMM, we can approximate *a posteriori* probability using the state equation as below:

$$\tilde{P} = P(O_1 \mid \boldsymbol{\mathcal{M}})P(O_2 \mid \boldsymbol{\mathcal{M}}) \cdots P(O_T \mid \boldsymbol{\mathcal{M}}) \sim P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}). \quad (2.64)$$

The validity of this approximation will be discussed in this and the next chapter.

As [6, 7, 8], we assume that there is a model $\boldsymbol{\mathcal{M}}$ with N states $q_i, i = 1, 2, \ldots, N$ and M discrete observation symbols k, $k = 1, 2, \ldots, M$. At each observation time t, we define the state probability vector $\boldsymbol{x}(t)$, and the observation probability vector, $\boldsymbol{y}(t)$, as follows:

$$\boldsymbol{x}'(t) \doteq (x_i(t), x_2(t), \ldots, x_N(t)) \quad (2.65)$$

$$\boldsymbol{y}'(t) \doteq (y_i(t), y_2(t), \ldots, y_M(t)) \quad (2.66)$$

where, $x_i(t)$ is the probability of being in q_i at discrete time t given the model $\boldsymbol{\mathcal{M}}$, $P(q_i \text{ at } t \mid \boldsymbol{\mathcal{M}})$, and $y_k(t)$ is the probability of generating symbol k at discrete time t given the model $\boldsymbol{\mathcal{M}}$, $P(k \text{ at } t \mid \boldsymbol{\mathcal{M}})$.

In these terms, the dynamics of the HMM are as follows:

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{u}(t)\delta(t) \quad (2.67)$$

$$\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{x}(t) \quad (2.68)$$

$$\tilde{P}(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}}) = \prod_{t=1}^{T} P(O_t \mid \boldsymbol{\mathcal{M}}) = \prod_{t=1}^{T} y_{O_t}(t) \quad (2.69)$$

where, $\boldsymbol{A}$ is the $N \times N$ state-transition matrix associated with the HMM whose (i,j) element, $a_{ji} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)$ for any t; $\boldsymbol{B}$ is the $M \times N$ observation probability matrix whose (k,j) element, $b_{kj} = P(k \mid q_j)$; and $\boldsymbol{u}(0)$ is some vector such that when $\boldsymbol{x}(0)$ is defined as zero, $\boldsymbol{x}(1)$ takes the proper initial values, with $\boldsymbol{u}(t)$ arbitrary but finite for all $t \neq 0$, and $\delta(t)$ is the Kronecker sequence. $y_{O_t}(t)$ corresponds to the k^{th} element of vector $\boldsymbol{y}(t)$. Here k is the symbol realized by O_t. $\tilde{P}(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})$ is the likelihood explained in the following.

The expressions (2.67)-(2.69) are not equivalent to (2.9)-(2.11) since the definitions of the state variables as well as the likelihoods from both sets of equations are different. State variable $\boldsymbol{\alpha}(t)$ is the joint probability of the partial observation sequence from an initial time to time t. However, $\boldsymbol{x}(t)$ is the probability of being in states at time t. Therefore, the likelihood $P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}})$ which needs to be evaluated from an initial time to a specific time t cannot be expressed by state variables $\boldsymbol{x}(t+1)$ and $\boldsymbol{y}$(t) without an independence condition that will be explained.

For reference, because $\boldsymbol{A}$ is a stochastic matrix, $\boldsymbol{x}(t)$ is a positive vector. In the Bakis structure of the HMM which is often employed in speech recognition, except a few initial values of t, $\boldsymbol{x}_i(t) \neq 0$ for any i. This implies that the system can be any one of states $[1, N]$ regardless of the preceding state.

From (2.67) and (2.68), we have

$$\boldsymbol{x}(t) = \boldsymbol{A}^{t-1}\boldsymbol{u}(0), \tag{2.70}$$

$$\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{A}^{t-1}\boldsymbol{u}(0). \tag{2.71}$$

These two probability values can be used to compute the state and observation probabilities at any time $t \in [1, T]$. The observation probability at time t for a observation

symbol O_t is

$$\begin{aligned} P(O_t \mid \mathcal{M}) &= y_{O_t}(t) \\ &= (b_1(O_t), b_2(O_t), \ldots, b_N(O_t)) \cdot \boldsymbol{x}(t) \\ &= b_1(O_t)x_1(t) + b_2(O_t)x_2(t) + \ldots + b_N(O_t)x_N(t) \qquad (2.72) \\ &= \mathbf{1}'\boldsymbol{\Delta}(t)\boldsymbol{x}(t). \end{aligned}$$

Here $\boldsymbol{b}'(O_t) = (b_1(O_t), b_2(O_t), \ldots, b_N(O_t))$. Note that

$$\boldsymbol{b}'(O_t) = \mathbf{1}'\boldsymbol{\Delta}(t). \qquad (2.73)$$

From (2.72), it is seen that the probability for a given observation at time t can be computed using state equation (2.70) and observation equation (2.71).

2.4 Transformations of State Equations

One of the merits of using the state-space structure developed above is that it admits the transformation of the state and observation equations into alternative formulations. Let us discuss this subject for the conventional F-B HMM and the TIA HMM.

2.4.1 Transformation of Time-Invariant State Equation

The original motivation for the time-invariant state equation was that the state-transition matrix could be diagonalized to significantly reduce the number of floating point operations required to compute the HMM likelihood[5] [4, 6, 7].

To diagonalize the state-transition matrix, let $\boldsymbol{x}(t) = \boldsymbol{M}\boldsymbol{z}(t)$. Where $\boldsymbol{M}$ is diagonalizing transformation on the state space. $\boldsymbol{M}$ is a square matrix with dimension

[5]In light of the remarks by Mitchell *et al.* [25], further discussion of this model appears in [4].

$N \times N$ and $\boldsymbol{z}(t)$ is a new state variable defined as $\boldsymbol{M}^{-1}\boldsymbol{x}(t)$ assuming that $\boldsymbol{M}^{-1}$ exists. If $\boldsymbol{A}$ does not have distinct eigenvalues other than zero, then $\boldsymbol{M}^{-1}$ does not exist. However, in the practical HMM application in speech processing, a speech signal is modeled as the result of random processes. $\boldsymbol{A}$ holds such variable transition information of random processes. Numerically, this leads that mostly the entries of $\boldsymbol{A}$ are different from each other. For example, the same phoneme spoken by different speakers will be acoustically different. Also, the same speaker may produce different versions of the same sound under different circumstances. Numerically, this leads that mostly the entries of $\boldsymbol{A}$ are quite different from each other and they do not have specific patterns such as singularity for the matrix for instance. Numerically, such diversity of realization of random processes for speech signals justifies assuming the existence of $\boldsymbol{M}^{-1}$ under a suitable size of number of state in the state model.

Equation (2.67) and (2.68) yield

$$\boldsymbol{M}\boldsymbol{z}(t+1) = \boldsymbol{A}\boldsymbol{M}\boldsymbol{z}(t) + \boldsymbol{u}(t)\delta(t) \tag{2.74}$$

$$\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{M}\boldsymbol{z}(t). \tag{2.75}$$

Diagonal dominance provides a relatively simple criterion for guaranteeing the nonsingularity of a matrix. An $N \times N$ real or complex matrix $\boldsymbol{A}$ is diagonally dominant if $|a_{i,i}| \geq \sum_{j \neq i} |a_{i,j}|, i = 1, \ldots, N$. $\boldsymbol{A}$ is also strictly diagonally dominant if strict inequality holds. If $\boldsymbol{A}$ is strictly diagonally dominant, then $\boldsymbol{A}$ is nonsingular [37]. Since $\boldsymbol{A}$ is nonsingular, a matrix $\boldsymbol{M}$, that is composed of a set of eigenvectors of $\boldsymbol{A}$, is nonsingular. Therefore, $\boldsymbol{M}^{-1}$ exists. From (2.74),

$$\boldsymbol{z}(t+1) = \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{M}\boldsymbol{z}(t) + \boldsymbol{M}^{-1}\boldsymbol{u}(t)\delta(t) \tag{2.76}$$

$$\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{M}\boldsymbol{z}(t). \tag{2.77}$$

Now suppose that $\boldsymbol{M} = \boldsymbol{PU}$, where $\boldsymbol{P}$ is the usual matrix of normalized eigenvectors and $\boldsymbol{U}$ is a special diagonal matrix such that the i^{th} element of the vector $\boldsymbol{U}^{-1}$ is the reciprocal of the i^{th} element of the vector $\boldsymbol{P}^{-1}u(0)$. As a consequence of this operation, each element of the excitation vector, $\boldsymbol{M}^{-1}\boldsymbol{u}(t)$, is unity at time zero.

It follows that

$$\begin{aligned} \boldsymbol{z}(t+1) &= \boldsymbol{U}^{-1}\boldsymbol{P}^{-1}\boldsymbol{APUz}(t) + \boldsymbol{U}^{-1}\boldsymbol{P}^{-1}\boldsymbol{u}(t)\delta(t) && (2.78) \\ &= \bar{\boldsymbol{A}}\boldsymbol{z}(t) + \bar{\boldsymbol{u}}(t)\delta(t) && (2.79) \\ \boldsymbol{y}(t) &= \boldsymbol{BPUz}(t) = \bar{\boldsymbol{B}}\boldsymbol{z}(t) && (2.80) \end{aligned}$$

where $\bar{\boldsymbol{A}} = \boldsymbol{U}^{-1}\boldsymbol{P}^{-1}\boldsymbol{APU}$ is a diagonal matrix, $\bar{\boldsymbol{u}}(t) = \boldsymbol{U}^{-1}\boldsymbol{P}^{-1}\boldsymbol{u}(t)$, and $\bar{\boldsymbol{B}} = \boldsymbol{BP}$. This result is significant because it separates all states into independent computations. Furthermore, this property provides a way to combine all HMMs in the system into one large state-space formulation [6, 7]. Moreover, as in (2.70) and (2.71),

$$\begin{aligned} \boldsymbol{z}(t) &= \bar{\boldsymbol{A}}^{t-1}\bar{\boldsymbol{u}}(0), && (2.81) \\ \boldsymbol{y}(t) &= \bar{\boldsymbol{B}}\bar{\boldsymbol{A}}^{t-1}\bar{\boldsymbol{u}}(0) && (2.82) \end{aligned}$$

where $\bar{\boldsymbol{A}}^{t-1}$ is computed easily.

In the HMM application to speech modeling, the Bakis condition is generally assumed, and, thus, $\boldsymbol{A}$ is a triangular matrix. Therefore, when there are K HMMs which need to be evaluated, it is necessary to compute eigensystems of K models. However, under the Bakis condition, all the eigenvalues are located on the diagonal positions of $\boldsymbol{A}$. Therefore, it is not necessary to compute eigenvalues. Furthermore, because of the strong diagonal property of $\boldsymbol{A}$, it is highly probable that there are quite a few cases across which eigenvalues can be shared. Hence practically we do not need to compute all the eigenvectors of K systems. Thus, computational load to

computing eigenvalues and eigenvectors of $\boldsymbol{A}$ among models can be possibly lessened by suitable preprocessing which examines the diagonal components of state-transition matrices across models.

2.4.2 Transformation of the Time-Varying State Equation

Let us return to the case of the time-varying state equations,

$$\begin{aligned} \boldsymbol{\alpha}(t+1) &= \boldsymbol{\Delta}(t+1)\boldsymbol{A}\boldsymbol{\alpha}(t) + \boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t) && (2.83) \\ y(t) &= P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) = \boldsymbol{C}'\boldsymbol{\alpha}(t). && (2.84) \end{aligned}$$

Let $\boldsymbol{\Lambda}$ and $\boldsymbol{P}$ denote the eigenvalue and eigenvector matrices of $\boldsymbol{A}$ respectively,

$$\boldsymbol{AP} = \boldsymbol{P\Lambda}. \tag{2.85}$$

Since $\boldsymbol{A}$ is nonsingular, $\boldsymbol{P}$ is nonsingular. Therefore

$$\begin{aligned} \boldsymbol{\alpha}(t+1) &= \boldsymbol{\Delta}(t+1)(\boldsymbol{P\Lambda P}^{-1})\boldsymbol{\alpha}(t) + \boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t) && (2.86) \\ &= \boldsymbol{\Delta}(t+1)\boldsymbol{P\Lambda P}^{-1}\boldsymbol{\alpha}(t) + \boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t) \\ \boldsymbol{P}^{-1}\boldsymbol{\alpha}(t+1) &= \boldsymbol{P}^{-1}\boldsymbol{\Delta}(t+1)\boldsymbol{P\Lambda P}^{-1}\boldsymbol{\alpha}(t) + \boldsymbol{P}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t). && (2.87) \end{aligned}$$

Now let

$$\boldsymbol{P}^{-1}\boldsymbol{\alpha}(t) = \bar{\boldsymbol{\alpha}}(t). \tag{2.88}$$

Then

$$\begin{aligned} \bar{\boldsymbol{\alpha}}(t+1) &= (\boldsymbol{P}^{-1}\boldsymbol{\Delta}(t+1)\boldsymbol{P\Lambda})\bar{\boldsymbol{\alpha}}(t) + (\boldsymbol{P}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi})\delta(t) && (2.89) \\ y(t) &= P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) = \boldsymbol{C}'\boldsymbol{P}\bar{\boldsymbol{\alpha}}(T). && (2.90) \end{aligned}$$

Since $\boldsymbol{P}$ is an eigenvector matrix of $\boldsymbol{A}$, this expression can be represented as follows:

$$\begin{aligned}
\bar{\boldsymbol{\alpha}}(t+1) &= \boldsymbol{P}^{-1}(\boldsymbol{AP}\boldsymbol{\Delta}(t+1) - \boldsymbol{AP}\boldsymbol{\Delta}(t+1) + \boldsymbol{\Delta}(t+1)\boldsymbol{AP})\bar{\boldsymbol{\alpha}}(t) \\
&\quad +(\boldsymbol{P}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi})\delta(t) \\
&= \boldsymbol{P}^{-1}\boldsymbol{AP}\boldsymbol{\Delta}(t+1)\bar{\boldsymbol{\alpha}}(t) + \boldsymbol{P}^{-1}(\boldsymbol{\Delta}(t+1)\boldsymbol{AP} - \boldsymbol{AP}\boldsymbol{\Delta}(t+1))\bar{\boldsymbol{\alpha}}(t) \\
&\quad +(\boldsymbol{P}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi})\delta(t) \\
&= \boldsymbol{\Lambda}\boldsymbol{\Delta}(t+1)\bar{\boldsymbol{\alpha}}(t) + \boldsymbol{P}^{-1}(\boldsymbol{\Delta}(t+1)\boldsymbol{P}\boldsymbol{\Lambda} - \boldsymbol{P}\boldsymbol{\Delta}(t+1)\boldsymbol{\Lambda})\bar{\boldsymbol{\alpha}}(t) \qquad (2.91) \\
&\quad +(\boldsymbol{P}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi})\delta(t) \\
&= \boldsymbol{\Lambda}\boldsymbol{\Delta}(t+1)\bar{\boldsymbol{\alpha}}(t) + \boldsymbol{P}^{-1}(\boldsymbol{\Delta}(t+1)\boldsymbol{P} - \boldsymbol{P}\boldsymbol{\Delta}(t+1))\boldsymbol{\Lambda}\bar{\boldsymbol{\alpha}}(t) \\
&\quad +(\boldsymbol{P}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi})\delta(t) \\
&= \boldsymbol{\Lambda}\boldsymbol{\Delta}(t+1)\bar{\boldsymbol{\alpha}}(t) + (\boldsymbol{P}^{-1}\boldsymbol{\Delta}(t+1)\boldsymbol{P} - \boldsymbol{\Delta}(t+1))\boldsymbol{\Lambda}\bar{\boldsymbol{\alpha}}(t) \\
&\quad +(\boldsymbol{P}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi})\delta(t),
\end{aligned}$$

where $\boldsymbol{\Lambda}\boldsymbol{\Delta}(t+1)$ is a diagonal matrix. To obtain a diagonalized state equation from (2.91), we need to have

$$\boldsymbol{P}^{-1}\boldsymbol{\Delta}(t+1)\boldsymbol{P} = \boldsymbol{\Delta}(t+1).$$

Or equivalently,

$$\boldsymbol{\Delta}(t+1)\boldsymbol{P} = \boldsymbol{P}\boldsymbol{\Delta}(t+1). \qquad (2.92)$$

If $\boldsymbol{P}$ has N distinct eigenvalues, the necessary and sufficient condition to satisfy the commutativity (2.92) is that all the eigenvectors of $\boldsymbol{P}$ should be same as those of $\boldsymbol{\Delta}(t+1)$ for all t [37]. However, $\boldsymbol{P}$ is not an eigenvector matrix of $\boldsymbol{\Delta}(t+1)$ but of $\boldsymbol{A}$. Therefore, $\boldsymbol{P}^{-1}\boldsymbol{\Delta}(t+1)\boldsymbol{P} \neq \boldsymbol{\Delta}(t+1)$ in general. Furthermore, $\boldsymbol{\Delta}(t)$ is time varying. Thus, a constant $\boldsymbol{P}$ which satisfies (2.92) for every t does not exist in

general. Therefore, there is no universal eigenvector matrix $\boldsymbol{P}$ which diagonalizes the time-varying F-B HMM.

In addition, consider the case in which the eigenvector matrix $\boldsymbol{P}$ changes with time, depending on $\boldsymbol{\Delta}(t)$. Let $\boldsymbol{P}_t$ denote a eigenvector matrix of $\boldsymbol{\Delta}(t)\boldsymbol{A}$. Then it follows that

$$\boldsymbol{\alpha}(t+1) = \boldsymbol{\Delta}(t+1)\boldsymbol{A}\boldsymbol{\alpha}(t) + \boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t) \tag{2.93}$$

$$\boldsymbol{P}_{t+1}^{-1}\boldsymbol{\alpha}(t+1) = \boldsymbol{P}_{t+1}^{-1}(\boldsymbol{\Delta}(t+1)\boldsymbol{A})\boldsymbol{P}_{t+1}\boldsymbol{P}_{t+1}^{-1}\boldsymbol{\alpha}(t) + \boldsymbol{P}_{t+1}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t) \tag{2.94}$$

$$\begin{aligned}\bar{\boldsymbol{\alpha}}(t+1) &= \overline{\boldsymbol{\Delta}(t+1)\boldsymbol{A}}\boldsymbol{P}_{t+1}^{-1}\boldsymbol{P}_t\boldsymbol{P}_t^{-1}\bar{\boldsymbol{\alpha}}(t) + \boldsymbol{P}_{t+1}^{-1}\boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t) \\ &= \overline{\boldsymbol{\Delta}(t+1)\boldsymbol{A}}(\boldsymbol{P}_{t+1}^{-1}\boldsymbol{P}_t)\bar{\boldsymbol{\alpha}}(t) + \boldsymbol{P}_{t+1}^{-1}(\boldsymbol{\Delta}(1)\boldsymbol{\pi})\delta(t)\end{aligned} \tag{2.95}$$

if $\boldsymbol{P}_t^{-1}$ exists for all t, where $\bar{\boldsymbol{\alpha}}(t+1) = \boldsymbol{P}_{t+1}^{-1}\boldsymbol{\alpha}(t+1)$, and $\boldsymbol{P}_{t+1}^{-1}(\boldsymbol{\Delta}(t+1)\boldsymbol{A})\boldsymbol{P}_{t+1} = \overline{\boldsymbol{\Delta}(t+1)\boldsymbol{A}}$. However, due to the sparseness of $\boldsymbol{\Delta}(t)$, $\boldsymbol{\Delta}(t)\boldsymbol{A}$ is singular most of the time and $\boldsymbol{P}_t^{-1}$ does not exist at these times. Additionally, even if $\boldsymbol{P}_t^{-1}$ exists for all t, it is necessary to compute $\boldsymbol{P}_t$ for each t, resulting in no computational benefit from the matrix diagonalization. Moreover, due to the fact that $\boldsymbol{P}_{t+1}^{-1}\boldsymbol{P}_t \neq \boldsymbol{I}$ in general, (2.95) implies that it is not possible to obtain a diagonalized state equation using the formulation above.

2.5 Analysis of Illegal Paths Caused by Approximation

In this section, we discuss the problem caused by the approximation of the F-B time-varying HMM by the TIA HMM. This issue was first noted by Mitchell *et al.* [25] following the publication of the original TIA HMM paper [7].

2.5.1 Likelihood Difference

We first discuss the relationship between $\tilde{P}$ of the TIA HMM and P of the F-B HMM. Those likelihood are significant for HMM evaluation in speech recognition.

As a matter of fact, the likelihood measure, $\tilde{P}(\mathcal{O} \mid \mathcal{M}) = \prod_{t=1}^{T} P(O_t \mid \mathcal{M})$ employed with state equations (2.67)-(2.69) is not linearly related to the *a posteriori* joint probability $P(\mathcal{O} \mid \mathcal{M}) = P(O_1, O_2, \ldots, O_T \mid \mathcal{M})$ used on the F-B HMM approach. For example, suppose there are three symbols $\{O_1, O_2, O_3\}$ in an observation string at times $t = 1, 2, 3$, respectively. Then by the "chain rule" of conditional probability,

$$P(O_1, O_2, O_3 \mid \mathcal{M}) = P(O_1 \mid \mathcal{M})P(O_2 \mid O_1, \mathcal{M})P(O_3 \mid O_2, O_1, \mathcal{M}) \quad (2.96)$$

which, if and only if O_1,O_2, and O_3 are conditionally independent[6], can be written as

$$\begin{aligned} P(O_1, O_2, O_3 \mid \mathcal{M}) &= P(O_1 \mid \mathcal{M})P(O_2 \mid \mathcal{M})P(O_3 \mid \mathcal{M}) && (2.97) \\ &= \tilde{P}(O_1, O_2, O_3 \mid \mathcal{M}). && (2.98) \end{aligned}$$

In this case, a time-invariant state equation can be applied to compute the "F-B" *a posteriori* probability $P(\mathcal{O} \mid \mathcal{M})$. However, the symbol occurrences are generally dependent upon the states in the HMM. The inequality of $P(\mathcal{O} \mid \mathcal{M})$ and $\tilde{P}(\mathcal{O} \mid \mathcal{M})$ caused by the assumption of independence among symbols without consideration of hidden state dependency was initially noted in [25], where a simple counter-example using a two-state HMM can be found[7].

To examine the differences in P and $\tilde{P}$ in more detail, consider a model with two states. From the F-B matrix formulation (2.13),

$$P(O_1, O_2, O_3 \mid \mathcal{M}) = \boldsymbol{C}' \boldsymbol{\Delta}(3) \boldsymbol{A} \boldsymbol{\Delta}(2) \boldsymbol{A} \boldsymbol{\Delta}(1) \boldsymbol{\pi} \quad (2.99)$$

[6]The dependent conditioning information being the state value.

[7]In light of the remarks by Mitchell *et al.* [25], further discussion of this model appears in [4].

$$
\begin{aligned}
&= \boldsymbol{C}' \begin{pmatrix} b_1(O_3) & 0 \\ 0 & b_2(O_3) \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \\
&\times \begin{pmatrix} b_1(O_2) & 0 \\ 0 & b_2(O_2) \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad (2.100) \\
&\times \begin{pmatrix} b_1(O_1) & 0 \\ 0 & b_2(O_1) \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}.
\end{aligned}
$$

Assume a left-to-right (Bakis) model so that $a_{12} = 0, \boldsymbol{C}' = (0,1)$ and $\boldsymbol{\pi}' = (1,0)$. Then,

$$
\begin{aligned}
P(O_1, O_2, O_3 \mid \boldsymbol{\mathcal{M}}) &= b_2(O_3)a_{21}b_1(O_2)a_{11}b_1(O_1) \\
&+ b_2(O_3)a_{22}b_2(O_2)a_{21}b_1(O_1). \qquad (2.101)
\end{aligned}
$$

On the other hand, the product of individual observation probabilities is

$$
\begin{aligned}
P(O_3 \mid \boldsymbol{\mathcal{M}})P(O_2 \mid \boldsymbol{\mathcal{M}})P(O_1 \mid \boldsymbol{\mathcal{M}}) &= \mathbf{1}'\boldsymbol{\Delta}(3) \begin{pmatrix} x_1(3) \\ x_2(3) \end{pmatrix} \mathbf{1}'\boldsymbol{\Delta}(2) \begin{pmatrix} x_1(2) \\ x_2(2) \end{pmatrix} \\
&\cdot \mathbf{1}'\boldsymbol{\Delta}(1) \begin{pmatrix} x_1(1) \\ x_2(1) \end{pmatrix} \qquad (2.102) \\
&= b_1(O_3)a_{11}^2 b_1(O_2)a_{11}b_1(O_1) \\
&+ b_1(O_3)a_{11}^2 b_2(O_2)a_{21}b_1(O_1) \\
&+ b_2(O_3)a_{21}a_{11}b_1(O_2)a_{11}b_1(O_1) \qquad (2.103) \\
&+ b_2(O_3)a_{21}a_{11}b_2(O_2)a_{21}b_1(O_1) \\
&+ b_2(O_3)a_{22}a_{21}b_1(O_2)a_{11}b_1(O_1) \\
&+ b_2(O_3)a_{22}a_{21}b_2(O_2)a_{21}b_1(O_1).
\end{aligned}
$$

Therefore, $\tilde{P}(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})$ involves extra cross terms which can be regarded as resulting

from one or more "illegal" state paths. Since all the terms of each $P(O_t \mid \mathcal{M})$ are multiplied together in computing $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$, this technique has been called an "anypath" method [4].

Although different from the F-B HMM likelihood, the "time-wise" $\tilde{P}(\mathcal{O} \mid \mathcal{M})$ probability has been useful in discovering new aspects of HMMs in this work. Moreover, the accompanying state equation is advantageous in that the resulting HMMs can be implemented to perform fast processing in real application with fewer resources than with the F-B HMM [6, 7, 25]. It is well-known that the training and evaluation of F-B HMMs are computationally very demanding [66]. To decrease the computational complexity, a few techniques have been proposed using vector-matrix formulations [27, 30]. However, since the proposed techniques are based on the time-varying F-B HMM, there is a limitation to the possible decrease in computational complexity. In spite of the apparent weakness of permitting illegal paths, the TIA HMM is a useful and effective model as we discuss later in this work.

2.5.2 Comparison of the State-Transition Matrices

Let us examine the state transitions of both F-B HMM and TIA HMM in more detail. Next, we discuss the discrepancy of the role of state-transition matrices of each model from the point of how to constitute available state paths of a speech utterance.

Since $\boldsymbol{\Delta}$ premultiplies the matrix $\boldsymbol{A}$ in the time-varying state-space HMM, diagonal elements of $\boldsymbol{\Delta}$ multiply the corresponding rows of $\boldsymbol{A}$. To examine the dynamics of the $\boldsymbol{\Delta A}$ matrix of the time-varying state equation, consider two two-state Bakis models and a test string $\mathcal{O} = \{O_1, O_2, \ldots, O_T\}$. At times t and $t+1$,

$$\boldsymbol{\alpha}(t) = \begin{pmatrix} b_1(O_t) & 0 \\ 0 & b_2(O_t) \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \boldsymbol{\alpha}(t-1) \tag{2.104}$$

$$\boldsymbol{\alpha}(t+1) = \begin{pmatrix} b_1(O_{t+1}) & 0 \\ 0 & b_2(O_{t+1}) \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \boldsymbol{\alpha}(t) \tag{2.105}$$

$$= \begin{pmatrix} b_1(O_{t+1}) & 0 \\ 0 & b_2(O_{t+1}) \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} b_1(O_t) & 0 \\ 0 & b_2(O_t) \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \boldsymbol{\alpha}(t-1). \tag{2.106}$$

Due to the Bakis condition, $a_{12} = 0$ and only transitions from state 1 to 2, 1 to 1, and 2 to 2 are legal. However, these legal jumps are also controlled by the probabilities in $\boldsymbol{\Delta}(t)$ and $\boldsymbol{\Delta}(t+1)$. Let $b_2(O_t) = b_1(O_{t+1}) = 1$ and $b_1(O_t) = b_2(O_{t+1}) = 0$ for instance. From these assumptions,

$$\boldsymbol{\alpha}(t+1) = \begin{pmatrix} a_{11} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ a_{21} & a_{22} \end{pmatrix} \boldsymbol{\alpha}(t-1) \tag{2.107}$$

$$= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \boldsymbol{\alpha}(t-1). \tag{2.108}$$

Therefore, the likelihood $P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})$ becomes zero. This is from the fact that by $\boldsymbol{\Delta}(t)$, the observation at t can be generated from state 2 and by $\boldsymbol{\Delta}(t+1)$, the observation at $t+1$ can be generated from state 1, but once an observation is generated by the second state, the path cannot return to state 1. Hence the matrix sequence $\boldsymbol{\Delta}(t)\boldsymbol{A}, t = 1, \ldots T$ inherently determines the allowable state paths depending on the elements of the observation string.

For the time-invariant model of (2.67) and (2.68), however, the computed likelihood (2.69) is an approximate value based on the assumption that O_t is uncondition-

ally independent of O_τ for all $\tau \in [1, t-1)$, meaning that

$$P(O_t \mid \mathcal{M}) = \sum_{i=1}^{N} P(O_t \mid q_t = i, \mathcal{M}) P(q_t = i \mid \mathcal{M}) \quad (2.109)$$

$$= \mathbf{1}' \boldsymbol{\Delta}(t) \boldsymbol{x}(t). \quad (2.110)$$

Here, algebraically the computation of $P(O_t \mid \mathcal{M})$ involves $\boldsymbol{x}(t)$ which is a state distribution that is dependent upon t only, not upon the history of the state path, nor of the symbol string. Thus, only the state-transition probabilities in $\boldsymbol{A}$ are responsible for predicting the state path in the TIA HMM. However, observation symbols play a significant role in deciding the feasibility, if not the value of the probability, of a state sequence. The viability of the TIA HMM depends on the degree to which probabilities computed for illegal state paths are small, rendering them infeasible. This will be discussed in the following chapters.

2.6 Validity of the Time-Invariant Approximation of the HMM

We examine the extent to which the TIA HMM represented by approximations (2.67)-(2.69) is a viable model for practical speech recognition. In particular, the relative significance of two significant matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the F-B HMM will be examined analytically and heuristically using several approaches. Of course, it is true that the following analysis are in fact heavily dependent upon the probabilities of $\boldsymbol{\Delta}(t)$.

2.6.1 Matrix Norm Approach

Let us first revisit two state-transition equations (2.11) and (2.25). Again, we have

$$P(O_1, O_2, \ldots, O_t, \ldots, O_T \mid \mathcal{M}) = \boldsymbol{C}' \boldsymbol{\alpha}(T) = \boldsymbol{C}' \boldsymbol{\Delta}(T) \boldsymbol{X}(T). \quad (2.111)$$

Note that the actual relationship between $\boldsymbol{\alpha}(t)$ and $\boldsymbol{X}(T)$ is given in (2.26). For $t \neq T$, the likelihood of the F-B HMM is

$$P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) \;=\; \mathbf{1}\boldsymbol{\alpha}(t) = \mathbf{1}\boldsymbol{\Delta}(t)\boldsymbol{X}(t). \tag{2.112}$$

For simplicity of analysis, consider an ergodic constraint on $\boldsymbol{A}$ so that all N states are legitimate final states. Then,

$$P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) \;=\; \boldsymbol{C}'\boldsymbol{\alpha}(t) = \boldsymbol{C}'\boldsymbol{\Delta}(t)\boldsymbol{X}(t) \tag{2.113}$$

for $t \in [1, T]$ with suitable initial conditions $\boldsymbol{\alpha}(1) = \Delta(1)\pi$ and $\boldsymbol{X}(1) = \pi$. Because $\boldsymbol{A}$ is a stochastic matrix, it is easily verified that

$$\boldsymbol{C}' \;=\; \boldsymbol{C}'\boldsymbol{I} = \boldsymbol{C}'\boldsymbol{A} = \boldsymbol{C}'\boldsymbol{A}^n \tag{2.114}$$

for any natural number n with a column vector $\boldsymbol{C}$ as defined in (2.12) and $\boldsymbol{I}$ defined as the identity matrix of suitable size. Then for any $1 \leq t \leq T$, *a posteriori* probability (2.112) can be represented variously as

$$\begin{aligned} P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) &= \boldsymbol{C}'\boldsymbol{\alpha}(t) \\ &= \|\boldsymbol{\alpha}(t)\|_1 \\ &= \boldsymbol{C}'\boldsymbol{A}\boldsymbol{\alpha}(t) \\ &= \|\boldsymbol{A}\boldsymbol{\alpha}(t)\|_1 \\ &= \boldsymbol{C}'\boldsymbol{\Delta}(t)\boldsymbol{X}(t) \\ &= \|\boldsymbol{\Delta}(t)\boldsymbol{X}(t)\|_1 \\ &= \boldsymbol{C}'\boldsymbol{A}\boldsymbol{\Delta}(t)\boldsymbol{X}(t) \\ &= \|\boldsymbol{A}\boldsymbol{\Delta}(t)\boldsymbol{X}(t)\|_1 \end{aligned} \tag{2.115}$$

$$= \|\boldsymbol{X}(t+1)\|_1, \tag{2.116}$$

where $\|\cdot\|$ represents the l_1 norm. If $\boldsymbol{\Omega}$ is a matrix such that

$$\boldsymbol{A}+\boldsymbol{\Omega} = \boldsymbol{I}, \tag{2.117}$$

then

$$\begin{aligned} \boldsymbol{C}'\boldsymbol{A} &= \boldsymbol{C}' = \boldsymbol{C}'\boldsymbol{I} \\ &= \boldsymbol{C}'(\boldsymbol{A}+\boldsymbol{\Omega}) \\ &= \boldsymbol{C}'\boldsymbol{A}+\boldsymbol{C}'\boldsymbol{\Omega}. \end{aligned} \tag{2.118}$$

Therefore, we have that

$$\boldsymbol{C}'\boldsymbol{\Omega} = \boldsymbol{C}'(\boldsymbol{\Omega})^n = \boldsymbol{0} \tag{2.119}$$

for any n with $\boldsymbol{C} = [1, 1, \ldots, 1]$. Here $\boldsymbol{0}$ is the zero vector with an appropriate dimension. The difference matrix $\boldsymbol{\Omega}$ indirectly points out the relative importance of the stochastic matrix $\boldsymbol{A}$ in the process of evaluation of the *a posteriori* probability for a given utterance. From (2.115),

$$\begin{aligned} \|\boldsymbol{\alpha}(t)\|_1 &= \|\boldsymbol{A}\boldsymbol{\alpha}(t)\|_1 = \|(\boldsymbol{I}-\boldsymbol{\Omega})\boldsymbol{\alpha}(t)\|_1 \\ &= \|\boldsymbol{\alpha}(t)-\boldsymbol{\Omega}\boldsymbol{\alpha}(t)\|_1. \end{aligned} \tag{2.120}$$

In particular, consider a diagonally dominant $\boldsymbol{A}$. For simplicity, let $\boldsymbol{A}$ is a 2×2

matrix and let ϵ_1 and ϵ_2 be two small numbers in the off-diagonal so that

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 1-\epsilon_1 & \epsilon_2 \\ \epsilon_1 & 1-\epsilon_2 \end{pmatrix}. \tag{2.121}$$

Also, let

$$\boldsymbol{\alpha}(t) = \begin{pmatrix} \alpha_1(t) \\ \alpha_2(t) \end{pmatrix}. \tag{2.122}$$

Then,

$$\boldsymbol{\Omega} = \begin{pmatrix} \epsilon_1 & -\epsilon_2 \\ -\epsilon_1 & \epsilon_2 \end{pmatrix}. \tag{2.123}$$

Therefore, all the elements of $\boldsymbol{\Omega}$ are composed of small numbers. From (2.121), we have that

$$\begin{aligned} \boldsymbol{A\alpha} &= \begin{pmatrix} 1-\epsilon_1 & \epsilon_2 \\ \epsilon_1 & 1-\epsilon_2 \end{pmatrix} \begin{pmatrix} \alpha_1(t) \\ \alpha_2(t) \end{pmatrix} \\ &= \begin{pmatrix} (1-\epsilon_1)\alpha_1(t) + \epsilon_2\alpha_2(t) \\ \epsilon_1\alpha_1(t) + (1-\epsilon_2)\alpha_2(t) \end{pmatrix}. \end{aligned} \tag{2.124}$$

If ϵ_1 and ϵ_2 are relatively small compared to the entries of $\boldsymbol{\alpha}$, (2.124) can be approximated as

$$\begin{aligned} \begin{pmatrix} (1-\epsilon_1)\alpha_1(t) + \epsilon_2\alpha_2(t) \\ \epsilon_1\alpha_1(t) + (1-\epsilon_2)\alpha_2(t) \end{pmatrix} &\approx \begin{pmatrix} (1-\epsilon_1)\alpha_1(t) \\ (1-\epsilon_2)\alpha_2(t) \end{pmatrix} \\ &= \begin{pmatrix} (1-\epsilon_1) & 0 \\ 0 & (1-\epsilon_2)\alpha_2(t) \end{pmatrix} \begin{pmatrix} \alpha_1(t) \\ \alpha_2(t) \end{pmatrix} \\ &= \boldsymbol{D\alpha}, \end{aligned} \tag{2.125}$$

where $\boldsymbol{D}$ is a diagonal matrix which is made up of diagonal elements only from matrix $\boldsymbol{A}$ so that $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{D}_1$. Therefore,

$$\begin{aligned} P(O_1, O_2, \ldots, O_t \mid \boldsymbol{\mathcal{M}}) &= \|\boldsymbol{\alpha}(t)\| \\ &= \|\boldsymbol{A}\boldsymbol{\alpha}(t)\| \\ &= \|(\boldsymbol{D} + \boldsymbol{D}_1)\boldsymbol{\alpha}(t)\| \\ &\approx \|\boldsymbol{D}\boldsymbol{\alpha}(t)\| \\ &= \boldsymbol{C}'\boldsymbol{D}\boldsymbol{\alpha}(t), \end{aligned} \tag{2.126}$$

where $\boldsymbol{D}$ is practically close to an identity matrix $\boldsymbol{I}$.

2.6.2 Likelihood Expansion Approach

The preceding discussion concerns figuring out the relative insignificance of stochastic matrix $\boldsymbol{A}$ at the final time t in light of the likelihood. In fact, however, we need to see the effect of $\boldsymbol{A}$ at each of the time instants $\{1, 2, \ldots, T\}$ at which a speech signal is evaluated.

To look at the influence of $\boldsymbol{A}$ matrix more closely, again consider a two-state model and the dynamics of the HMM over a few time instants. The results of this analysis can be extended to any size state model. Let $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ be two HMMs for speech evaluation. For simplicity, let us suppose that an observation string is composed of three symbols $\{O_1, O_2, O_3\}$. Then, we need to evaluate the final two likelihoods,

$$P(O_1, O_2, O_3 \mid \boldsymbol{\mathcal{M}}_1) = (1,1)\boldsymbol{\Delta}_1(3)\boldsymbol{A}_1\boldsymbol{\Delta}_1(2)\boldsymbol{A}_1\boldsymbol{\Delta}_1(1)\boldsymbol{\pi}_1 \tag{2.127}$$

$$P(O_1, O_2, O_3 \mid \boldsymbol{\mathcal{M}}_2) = (1,1)\boldsymbol{\Delta}_2(3)\boldsymbol{A}_2\boldsymbol{\Delta}_2(2)\boldsymbol{A}_2\boldsymbol{\Delta}_2(1)\boldsymbol{\pi}_2, \tag{2.128}$$

where subscripts 1 and 2 denote $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ respectively. Let us assume a Bakis structure for $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. Then $a_{1,12} = a_{1,22} = a_{2,12} = a_{2,22} = 0$. Like the notation of

$\boldsymbol{A}_1, \boldsymbol{A}_2$, the subscript of n from $a_{n,ji}$ denotes a transition probability from state i to j in HMM n. So, the likelihoods for $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ become

$$\begin{aligned} P(O_1, O_2, O_3 \mid \boldsymbol{\mathcal{M}}_1) &= b_{1,1}(3)a_{1,11}b_{1,1}(2)a_{1,11}b_{1,1}(1)\pi_{1,1} \\ &+ b_{1,2}(3)a_{1,21}b_{1,1}(2)a_{1,11}b_{1,1}(1)\pi_{1,1} \\ &+ b_{1,2}(3)a_{1,22}b_{1,2}(2)a_{1,21}b_{1,1}(1)\pi_{1,1} \end{aligned} \tag{2.129}$$

$$\begin{aligned} P(O_1, O_2, O_3 \mid \boldsymbol{\mathcal{M}}_2) &= b_{2,1}(3)a_{2,11}b_{2,1}(2)a_{2,11}b_{2,1}(1)\pi_{2,1} \\ &+ b_{2,2}(3)a_{2,21}b_{2,1}(2)a_{2,11}b_{2,1}(1)\pi_{2,1} \\ &+ b_{2,2}(3)a_{2,22}b_{2,2}(2)a_{2,21}b_{2,1}(1)\pi_{2,1}. \end{aligned} \tag{2.130}$$

Here similarly to $a_{n,ji}$, $b_{n,j}(k)$ denotes an observation probability for symbol k from state j in HMM n and $\pi_{n,i}$ denotes a initial state probability for a state i in HMM n. In (2.129), if $\boldsymbol{A}_1$ is close to $\boldsymbol{A}_2$, then $\boldsymbol{B}$ plays a decisive role in computing the likelihoods. It is difficult to show analytically how much the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ affect the likelihood in general since its result differs depending on the time-varying input speech signals. However, it is possible roughly to estimate the relative contribution of each matrix.

2.6.3 Matrix Inversion Approach

Here we consider the application of the vector-matrix formulation of the HMM to assess the relative importance of $\boldsymbol{A}$ and $\boldsymbol{B}$ to the likelihood.

Previously, a matrix norm has been applied for an ergodic model to obtain a closed form for the *a posteriori* probability quantity of a given speech utterance at a specific time as (2.115) or (2.116). Now consider a general case covering all time indices.

Reconsider a model with a time-varying state-equation representation as (2.22).

Multiplying both sides of (2.22) by $\boldsymbol{A}^{-1}$,

$$\boldsymbol{A}^{-1}\boldsymbol{X}(t+1) = \boldsymbol{\Delta}(t)\boldsymbol{X}(t) + \boldsymbol{A}^{-1}\boldsymbol{\pi}\delta(t). \tag{2.131}$$

Let us assume a Bakis structure with strong diagonal elements in $\boldsymbol{A}$. As before, let $\boldsymbol{\Omega} = \boldsymbol{I} - \boldsymbol{A}$. Then

$$(\boldsymbol{I} - \boldsymbol{\Omega})^{-1}\boldsymbol{X}(t+1) = \boldsymbol{\Delta}(t)\boldsymbol{X}(t) + \boldsymbol{A}^{-1}\boldsymbol{\pi}\delta(t). \tag{2.132}$$

Further

$$(\boldsymbol{I} - \boldsymbol{\Omega})^{-1} = \boldsymbol{I} + \boldsymbol{\Omega} + \boldsymbol{\Omega}^2 + \boldsymbol{\Omega}^3 + \ldots \tag{2.133}$$

$$= \boldsymbol{I} + \sum_{n=1}^{\infty} \boldsymbol{\Omega}^n. \tag{2.134}$$

Therefore,

$$(\boldsymbol{I} + \sum_{n=1}^{\infty} \boldsymbol{\Omega}^n)\boldsymbol{X}(t+1) = \boldsymbol{\Delta}(t)\boldsymbol{X}(t) + \boldsymbol{A}^{-1}\boldsymbol{\pi}\delta(t) \tag{2.135}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{\Delta}(t)\boldsymbol{X}(t) - (\sum_{n=1}^{\infty} \boldsymbol{\Omega}^n)\boldsymbol{X}(t+1) + \boldsymbol{A}^{-1}\boldsymbol{\pi}\delta(t). \tag{2.136}$$

From the definition of $\boldsymbol{\Omega}$ and (2.134),

$$\sum_{n=1}^{\infty} \boldsymbol{\Omega}^n = \boldsymbol{A}^{-1} - \boldsymbol{I}. \tag{2.137}$$

Substituting in (2.136),

$$\boldsymbol{X}(t+1) = \boldsymbol{\Delta}(t)\boldsymbol{X}(t) - (\boldsymbol{A}^{-1} - \boldsymbol{I})\boldsymbol{X}(t+1) + \boldsymbol{A}^{-1}\boldsymbol{\pi}\delta(t) \tag{2.138}$$

$$= \hat{\boldsymbol{X}}(t) + \tilde{\boldsymbol{X}}(t+1) + \boldsymbol{A}^{-1}\boldsymbol{\pi}\delta(t) \tag{2.139}$$

where $\hat{\boldsymbol{X}}(t) = \boldsymbol{\Delta}(t)\boldsymbol{X}(t)$ and $\tilde{\boldsymbol{X}}(t+1) = -(\boldsymbol{A}^{-1} - \boldsymbol{I})\boldsymbol{X}(t+1)$. Because $\boldsymbol{A}$ is assumed to have strong diagonal elements, $\boldsymbol{X}(t+1) = \hat{\boldsymbol{X}}(t)$ for all $t > 0$. To see $\boldsymbol{X}(t+1)$ quantitatively, consider, for example, a simple case in which $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$,

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \tag{2.140}$$

Then,

$$\boldsymbol{A}^{-1} - \boldsymbol{I} = \frac{1}{\det(\boldsymbol{A})} \cdot \begin{pmatrix} a_{22} - a_{11}a_{22} + a_{12}a_{21} & -a_{12} \\ -a_{21} & a_{11} - a_{11}a_{22} + a_{12}a_{21} \end{pmatrix} \tag{2.141}$$

assuming that $\det(\boldsymbol{A}) = a_{11}a_{22} - a_{12}a_{21} > 0$ which is justified in the present analysis because of the strong diagonal property. Contrary to the denominator, all entries of the numerator of (2.141) are small numbers. For the Bakis model, $a_{22} = 1$ and $a_{12} = 0$, so that

$$\boldsymbol{A}^{-1} - \boldsymbol{I} = \begin{pmatrix} \frac{a_{21}}{a_{11}} & 0 \\ \frac{-a_{21}}{a_{11}} & 0 \end{pmatrix}. \tag{2.142}$$

Accordingly, $\|(\boldsymbol{A}^{-1} - \boldsymbol{I})\boldsymbol{X}(t+1)\|$ is very small compared with the values in $\boldsymbol{X}(t)$. For any size of N, we reach the same conclusion. This is further support for the notion that observation probabilities are much more significant than the state-transition probabilities in computing the likelihood.

2.6.4 Eigenanalysis Approach

Let us examine the eigensystem of the state-transition matrix of the time-varying F-B HMM. This approach also leads to the conclusion that $\boldsymbol{A}$ is relatively insignificant compared to $\boldsymbol{\Delta}$ in light of the likelihood measure.

From (2.86), the diagonalization of time-varying F-B HMM is explicitly given by

$$\boldsymbol{\alpha}(t+1) = \boldsymbol{\Delta}(t+1)\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1}\boldsymbol{\alpha}(t) + \boldsymbol{\Delta}(1)\boldsymbol{\pi}\delta(t). \tag{2.143}$$

Since $\boldsymbol{\Delta}(t+1)$ is a diagonal matrix, analyzing $\boldsymbol{A}$ (or $\boldsymbol{P}$) is sufficient. Note that generally the operation of vector-matrix-vector multiplication (2.90) to compute the likelihood does not have a straightforward relation to the eigensystem of the matrix. However, consider the condition in which $\boldsymbol{A}$ is close to the identity matrix $\boldsymbol{I}$.

To observe the significance of $\boldsymbol{A}$, consider two HMMs which are numerically very close to each other. Let $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ be state-transition matrix of $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. Let $\boldsymbol{A}_1$ have the Bakis topology. For analysis purpose, suppose that all the entries of $\boldsymbol{A}_2$ are close to those of $\boldsymbol{A}_1$ and they are close to $\boldsymbol{I}$ so that they have strong diagonal property. Further assume that the other two matrices $\boldsymbol{B}_1$ and $\boldsymbol{\pi}_1$ from $\mathcal{M}_1$ are the same as $\boldsymbol{B}_2$ and $\boldsymbol{\pi}_2$ from $\mathcal{M}_2$. Then let us estimate the likelihood from $\boldsymbol{A}_2$ in terms of $\boldsymbol{A}_1$.

Consider a practical case first in which $\boldsymbol{A}_2$ which is close to $\boldsymbol{A}_1$ is of the Bakis topology so that the diagonal entries of $\boldsymbol{A}_2$ are eigenvalues themselves. During matrix multiplications for likelihood computation, the eigenvalues of the state-transition matrix are explicitly involved in the matrix operations as explicit (diagonal) entries in the matrix. In this case, it is trivial since both $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ produce the similar likelihood.

Next consider the case such that $\boldsymbol{A}_2$ is not of the Bakis topology but it is still strongly diagonally dominant. The non-Bakis condition makes the analysis difficult since the eigenstructure of the system varies depending on changes in the entries of the matrix. Because $\boldsymbol{A}_2$ is no longer triangular, the eigenvalues of the matrix are not explicitly involved in the matrix multiplication. However, the Gerschgorin circle theorem [37, 38, 39] provides another way to assess the matrix multiplication

approximately. The theorem tells the possible relative locations of corresponding eigenvalues after the entries of a matrix changes a little from an original matrix. To apply the theorem, let λ be an eigenvalue of the $N \times N$ matrix $\boldsymbol{A}_2$. Then, by the Gerschgorin theorem, each eigenvalue lies in at least one of the discs with center $a_{1,ii}$ and radius $r_i = \sum_{j \neq i} |a_{1,ij}|, \quad i = 1, \ldots, N$ in the complex plane,

$$|\lambda - a_{1,ii}| \leq r_i. \tag{2.144}$$

Because $\boldsymbol{A}_2$ is diagonally dominant, the size of each Gerschgorin disk is very small. Moreover, since $\boldsymbol{A}_2$ is a stochastic matrix, one eigenvalue remains unity. For matrix computation, consider eigenvectors of $\boldsymbol{A}_2$. Although the eigenvalues of $\boldsymbol{A}_2$ are not affected much and the Euclidean distances between respective eigenvalues of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are small, the sensitivity of eigenvectors depends on the eigenvalue sensitivity and separation. In particular, in case of identical eigenvalues, there exists an infinite set of possible eigenvectors because of linear dependency. Therefore, the eigenvector conditions are not helpful to estimate the likelihood with matrix $\boldsymbol{A}_2$ in association with $\boldsymbol{A}_1$.

However, in case that $\boldsymbol{\Delta}(t)$ is sparse[8] so that the effect of off-diagonal elements are reduced, we may get likelihood results with $\boldsymbol{A}_2$ which are close to those with $\boldsymbol{A}_1$. The results are in fact mostly dependent upon the probabilities of $\boldsymbol{\Delta}(t)$.

For the TIA HMM, (2.76) and (2.77), eigenanalysis approach is better applicable than for the F-B HMM because the time-invariant state equation does not depend on the observation symbol string.

[8]The sparseness of $\boldsymbol{B}$ will be explained in Chap. 3.

Chapter 3

Practical Issues in the Use of the TIA HMM

This chapter is devoted to the practical issues related to the TIA HMM. Here, the "practical issues" relate principally to the problem of illegal state sequences in the TIA HMM. Additionally, the technique proposed by Turin [27] to reduce the computational load for likelihood computation will be reexamined. Then, a new approach will be proposed and derived to reduce the computational work for likelihood computation based on a condition imposed on an utterance by Turin. Through such an approach, we will show that we can obtain more computational savings as well as fast evaluation with reduced computational resources. Even though such an assumption imposed by Turin on a speech utterance is not ubiquitous in real speech signals, however, this study will be significant to compare the efficiency of computational savings of the new approach with that of Turin. We will discuss the problem related to computational savings of HMMs first in the following section.

3.1 Efficient Evaluation Technique

In [27], Turin suggests a new technique for computing the HMM likelihood efficiently using a vector-matrix formulation of the HMM. Turin's approach assumes that the observation string extracted from the speech has long stretches of identical observations. Let us discuss and examine his method so as to derive more computational saving technique.

Before presenting a new technique, we briefly review the Turin's method [27]. Assume that the observation string has long stretches of identical observations, say,

$$O_{t+1} = O_{t+2} = \cdots = O_{t+r} \tag{3.1}$$

with r-repetitions of the symbol. When there are many blocks of repetitive strings, the following development can be reapplied. From (2.13), the likelihood is given by

$$\begin{aligned} P(O_1, O_2, \ldots, O_T \mid \mathcal{M}) &= \boldsymbol{C}' \boldsymbol{\Delta}(T) \boldsymbol{A} \boldsymbol{\Delta}(T-1) \boldsymbol{A} \cdots \boldsymbol{\Delta}(2) \boldsymbol{A} \boldsymbol{\Delta}(1) \boldsymbol{\pi} \qquad (3.2) \\ &= \boldsymbol{C}' \boldsymbol{\Delta}(T) \boldsymbol{A} \boldsymbol{\Delta}(T-1) \boldsymbol{A} \cdots \boldsymbol{\Delta}(t+r+1) \boldsymbol{A} \\ &\quad (\boldsymbol{\Delta}(t+1)) \boldsymbol{A})^r \boldsymbol{\Delta}(t) \boldsymbol{A} \cdots \boldsymbol{\Delta}(2) \boldsymbol{A} \boldsymbol{\Delta}(1) \boldsymbol{\pi} \qquad (3.3) \end{aligned}$$

under (3.1). Thus, the problem becomes how to compute a matrix $(\boldsymbol{\Delta}(t+1))\boldsymbol{A})^r$ efficiently. Among several algorithms proposed by Turin [27] for computing $(\boldsymbol{\Delta}(t+1))\boldsymbol{A})^r$ one of the technique suggested using [28] is as follows:

1. Let $\{b_{k-1} b_{k-1} \cdots b_1 b_0\}$ be a binary representation of r as

$$r = b_0 + 2b_1 + \ldots + 2^{k-1} b_{k-1}. \tag{3.4}$$

Also, let

$$Q_0 = \boldsymbol{I} \tag{3.5}$$

$$R_1 = \boldsymbol{\Delta}(t+1)\boldsymbol{A}. \tag{3.6}$$

2. For $i = 1, \ldots, k+1$,

$$R_{i+1} = R_i^2 \tag{3.7}$$

$$Q_i = \begin{cases} Q_{i-1} & \text{if } b_{i-1} = 0 \\ Q_{i-1}R_i & \text{if } b_{i-1} = 1. \end{cases} \tag{3.8}$$

3. Termination

$$(\boldsymbol{\Delta}(t+1))\boldsymbol{A})^r = Q_k \tag{3.9}$$

This algorithm requires on the average $3N^3 \log_{10} r$ floating-point operations (flops) in calculating $(\boldsymbol{\Delta}(t+1))\boldsymbol{A})^r$. It is obvious that we get more computational savings when for r is large. Depending on r, however, the computational savings varies. For example, when $r = 2^n$ for $n \in \mathbb{N}$, the large computational savings can be obtained. On the other hand, the computational savings becomes relatively small when $r = 2^n - 1$. As well as such variability of computational savings, this algorithm still requires a recursive squaring of $(\boldsymbol{\Delta}(t+1)\boldsymbol{A})$.

To improve upon techniques proposed for computing $(\boldsymbol{\Delta}(t+1)\boldsymbol{A})^r$ [27, 28], we develop a more computationally efficient technique for computing this repetitive matrix multiplication based on a linear transformation of the matrix. This method is particularly efficient in cases where the matrix R_1 from (3.6) is a sparse, near-triangular matrix, typical of the HMM structure. The derivation follows.

In Chapter 2, a similarity transformation of the non-singular matrix $\boldsymbol{A}$ was used

to obtain more computational savings in the TIA HMM. Similarly, a linear transformation can be applied to the computation of $(\boldsymbol{\Delta}(t+1)\boldsymbol{A})^r$. In this case, this matrix product is singular most of the time.

Initially, suppose that $\boldsymbol{\Delta}(t+1)$ is non-singular. Then, the resulting product $\boldsymbol{\Delta}(t+1)\boldsymbol{A}$ is non-singular; thus, the matrix $(\boldsymbol{\Delta}(t+1)\boldsymbol{A})$ can be expressed as the product of three matrices,

$$\boldsymbol{\Delta}(t+1)\boldsymbol{A} = \boldsymbol{P}(t+1)\boldsymbol{D}(t+1)\boldsymbol{P}^{-1}(t+1), \tag{3.10}$$

where $\boldsymbol{P}(t+1)$ is an eigenvector matrix of, and $\boldsymbol{D}(t+1)$ is a diagonal eigenvalue matrix of, $\boldsymbol{\Delta}(t+1)\boldsymbol{A}$. Therefore,

$$(\boldsymbol{\Delta}(t+1)\boldsymbol{A})^r = \boldsymbol{P}(t+1)\boldsymbol{D}^r(t+1)\boldsymbol{P}^{-1}(t+1). \tag{3.11}$$

Since $\boldsymbol{D}(t+1)$ is a diagonal matrix, computing $\boldsymbol{D}^r(t+1)$ is straightforward. If $\boldsymbol{D}(t+1)$ is $N \times N$, for example, it takes only $N \times r$ flops to compute $\boldsymbol{D}^r(t+1)$. Likewise, computing $\boldsymbol{P}^{-1}(t+1)$ from $\boldsymbol{P}(t+1)$ takes N^3 flops. This is not computationally demanding when N is not large. N is practically not over 6 in HMMs.

Second, suppose that $\boldsymbol{\Delta}(t+1)\boldsymbol{A}$ is singular because of zero elements in the $\boldsymbol{\Delta}(t+1)$ matrix. Note that $\boldsymbol{A}$ is a non-singular matrix. In this case, we still can choose non-singular eigenmatrix $\boldsymbol{P}(t+1)$ because it is possible to have any linearly independent eigenvector corresponding to a zero eigenvalue. Thus, a non-singular matrix $\boldsymbol{P}(t+1)$ exists always regardless of values of $\boldsymbol{\Delta}(t+1)$. Hence, there is always a valid relation (3.11).

To compare the required number of floating operations for computing $(\boldsymbol{\Delta}(t+1)\boldsymbol{A})^r$ between three techniques described above, consider a simple case as follows. Suppose that all the diagonal entries of $\boldsymbol{\Delta}(t+1)$ are not zero, and $\boldsymbol{A}$ is a triangular matrix with allowing any forward state jump. Furthermore, assume that state transition matrix

$\boldsymbol{\Delta}(t+1)\boldsymbol{A}$ has distinct eigenvalues. Then, the necessary computational complexities for three techniques are shown in Table 3.1. For example, when $r = 10$ and $N = 5$,

Approaches	flops
Conventional F-B HMM	$\frac{N}{2}(N+1)r + \frac{N}{6}(N+1)(N+2)(r-1)$
Turin's algorithm	$3N^3 \log_{10} r$
Similarity Transformation	$N(N+1) + Nr + N^2 + N^3 + \frac{N}{6}(N+1)(N+2)$

Table 3.1: Approximate computational complexities for computing $(\boldsymbol{\Delta}(t+1)\boldsymbol{A})^r$ by three different approaches.

the approximate complexities for computing $(\boldsymbol{\Delta}(t+1)\boldsymbol{A})^r$ are 2040 flops by the conventional scalar recursive F-B HMM algorithm, 375 flops by Turin's algorithm, and 265 flops by the similarity transformation in (3.11).

It is obvious that, like [27], savings in computing P increases with increase of r. In contrast to the technique in [27], however, the necessary load for computing $(\boldsymbol{\Delta}(t+1))\boldsymbol{A})^r$ using the similarity transformation is less sensitive to r since the computational load increases proportionally with rate of N. On the other hand, for the Turin's algorithm, it increases with $3N^3$ associated with $\log_{10} r$.

Now, consider the TIA HMM under the Turin's assumption that the observation string has long stretches of identical observations. From (2.78) and (2.80), the partial likelihood from $t+1$ to $t+r$ becomes

$$\prod_{\tau=t+1}^{t+r} P(O_\tau \mid \mathcal{M}) = \prod_{\tau=t+1}^{t+r} \{\bar{\boldsymbol{b}}'(O_\tau)\boldsymbol{z}(\tau)\} \tag{3.12}$$

$$= \prod_{\tau=t+1}^{t+r} \{\bar{\boldsymbol{b}}'(O_\tau)\bar{\boldsymbol{A}}\boldsymbol{z}(\tau-1)\} \tag{3.13}$$

$$= (\bar{\boldsymbol{b}}'(O_{t+1})\bar{\boldsymbol{A}}^r\boldsymbol{z}(t))(\bar{\boldsymbol{b}}'(O_{t+1})\bar{\boldsymbol{A}}^{r-1}\boldsymbol{z}(t))$$

$$\cdots\ (\bar{\boldsymbol{b}}'(O_{t+1})\bar{\boldsymbol{A}}^2\boldsymbol{z}(t))(\bar{\boldsymbol{b}}'(O_{t+1})\bar{\boldsymbol{A}}\boldsymbol{z}(t)) \tag{3.14}$$

because $\bar{\boldsymbol{b}}'(O_\tau)$ and $\boldsymbol{z}(\tau)$ do not change over $t+1 \leq \tau \leq t+r$. In (3.14), however,

it is not possible to get further computational savings for $\prod_{\tau=t+1}^{t+r} P(O_\tau \mid \mathcal{M})$ even though $\bar{\boldsymbol{A}}$ is a diagonal matrix, because $\bar{\boldsymbol{A}}$ is between row and column vectors which form a dot product.

3.2 Analysis of the TIA HMM

In this section, the structure of the TIA HMM will be focused on in detail.

3.2.1 Likelihood Structures

The HMM is obviously based on the assumption that at each time, a symbol is generated as a consequence of a state transition or result of state entrance. Depending on the type of the HMM, a symbol is modeled to be generated either during or after state jump [4]. In either case, the likelihood is made up of T sequential multiplications of pairs of a_{ji} from $\boldsymbol{A}$ and $b_j(k)$ from $\boldsymbol{B}$ for a T-length speech utterance. The initial state probability π_j is a special case of a_{ji} which could be represented symbolically as $\pi_i = a_{i0}$.

For simplicity, assume that there is a single legal state path. Then, according to the dynamics of the HMM, the formulation of the likelihood is as follows:

$$P(\boldsymbol{\mathcal{O}} \mid \mathcal{M}) = \{a_{q_1,0} b_{q_1}(O_t)\} \cdots \{a_{q_t,q_{t-1}} b_{q_t}(O_t)\}\{a_{q_{t+1},q_t} b_{t+1}(O_{t+1})\} \cdots . \tag{3.15}$$

In case there are more than a single legal state path, the sum of terms of form (3.15) comprises the overall likelihood. Therefore, regardless of the number of legal states, the number of a_{ji}s are the same as that of b_js in the likelihood equation when evaluated by the F-B HMM or Viterbi HMM.

For the TIA HMM, however, $x_i(t)$ is computed from Markov process (2.67); thus, $x_i(t)$ is sum of terms composed of a_{ji} and takes a "sum-of-products" formulation. In

proportion to t, the exponent of a_{ji} increases and $x_i(t)$ could be up to $(t-1)^{st}$ powers of a_{ji}. Therefore, $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$ requires T terms of (2.72) and thus it takes on "product-of-sums" formulation. Also, in $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$, $(t-1)^{st}$ powers of a_{ji} is multiplied by a single $b_j(O_t)$ rather just a_{ji}.

In an extreme case, for example, suppose that only a single element of a vector $\boldsymbol{b}' = (b_1(O_t), b_2(O_t), \ldots, b_N(O_t))$ is not zero for $t \in [1, T]$. Such a condition is rarely satisfied for real speech signals under a HMM framework since it implies a simple "non-hidden" Markov model. Practically, however, for a few $t \in [1, T]$, such phenomenon occurs frequently in the F-B HMM training. It is not easy to quantify for how many times the assumption is fit because the length of training speech utterance varies and it really data-dependent. Moreover, initial values of $\boldsymbol{A}$ and $\boldsymbol{B}$ randomly assigned influence the estimated values of $\boldsymbol{A}$ and $\boldsymbol{B}$ during training.

The case that only one element of $\boldsymbol{b}'$ has non-zero probability in training is investigated with fifteen different initial settings for $\boldsymbol{A}$ and $\boldsymbol{B}$. As before, a Bakis constraint is considered. Also, fifteen training utterance of word "six" and "four" are used for this simulation. For the word "six" with 5-state HMM, the rate of the case that only a single element of $\boldsymbol{b}'$ has non-zero probability is 28.5% of T-observations in training sequences. On the other hand, for the word "four" with 3-state HMM, the rate reaches 84% of T-observations in training sequences.

According to the assumption, we have

$$P(O_t \mid \mathcal{M}) \;=\; b_i(O_t) x_i(t) \tag{3.16}$$

from (2.72). Here, i designates the state that produces O_t. Let $N = 2$ for instance, and consider a case with the Bakis constraint on $\boldsymbol{A}$. It follows that

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}(1) \;=\; \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{3.17}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}(2) = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix}, \tag{3.18}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}(3) = \begin{pmatrix} a_{11}^2 \\ a_{21}a_{11} + a_{22}a_{21} \end{pmatrix}, \tag{3.19}$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}(4) = \begin{pmatrix} a_{11}^3 \\ a_{21}a_{11}^2 + a_{22}a_{21}a_{11} + a_{22}^2a_{21} + a_{22}a_{21}a_{11} \end{pmatrix} \tag{3.20}$$

$$\vdots \quad \vdots \quad \ddots$$

If $P = P(O_1 \mid \mathcal{M})P(O_2 \mid \mathcal{M})P(O_3 \mid \mathcal{M})P(O_4 \mid \mathcal{M})$ is to be evaluated for the observation string $\mathcal{O} = \{O_1, O_2, O_3, O_4\}$ for instance, then the likelihood becomes

$$P = P(O_1 \mid \mathcal{M})P(O_2 \mid \mathcal{M})P(O_3 \mid \mathcal{M})P(O_4 \mid \mathcal{M}) \tag{3.21}$$

$$= b_{q_1}(O_1)x_{q_1}(1)b_{q_2}(O_2)x_{q_2}(2)b_{q_3}(O_3)x_{q_3}(3)b_{q_4}(O_4)x_{q_4}(4), \tag{3.22}$$

where $\{q_1, q_2, q_3, q_4\}$ is a sequence of states which produce a symbol string $\mathcal{O}$. In case $\{q_1 = 1, q_2 = 1, q_3 = 2, q_4 = 2\}$, for example, which is one of the legal state sequences that can produce a symbol sequence $\mathcal{O}$,

$$P = \prod_{t=1}^{4} P(O_t) \tag{3.23}$$

$$= \{\pi_1 b_1(O_1)b_1(O_2)b_2(O_3)b_2(O_4)\}$$

$$\cdot \{a_{11}(a_{21}a_{11} + a_{22}a_{21})(a_{21}a_{11}^2 + a_{22}a_{21}a_{11} + a_{22}^2a_{21} + a_{22}a_{21}a_{11})\}. \tag{3.24}$$

The highest order of the polynomial in the a_{ji} coefficients is six, and these polynomials are multiplied by $b_1(O_1)b_1(O_2)b_2(O_3)b_2(O_4)\pi_1$. These products are not consistent with (3.15) in the sense of Moore or Mealy forms of HMMs.

Comparing (3.15) with (3.22), roughly speaking, a_{ji} in (3.15) is substituted for x_{q_t} in (3.22). With "extra" polynomials composed of a_{ji}s in x_{q_t}, the likelihood $\prod_{t=1}^{T} P(O_t \mid$

$\mathcal{M}$) is different from $P(\mathcal{O} \mid \mathcal{M})$ of the F-B HMM. It is not simple to figure out quantitatively how much such extra likelihood from the TIA HMM affects the overall performance of the speech recognition system. The performance is data-dependent and dependent as well on the values of $a_{ji}, b_j(O_t)$.

Even though two likelihood measures are not identical, the performance of a speech recognition system using $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$ is not significantly degraded. This fact does not imply that state-transition probabilities are not informative. They are important to constitute a possible state sequence as well as the likelihood in the HMM. In the F-B HMM, they are vital. In case of the F-B HMM, the state probabilities themselves regulate the state path to some extent. In addition, with such state probabilities made up of state-transition probabilities x_{q_t}s instead of state-transition probabilities a_{ji}s, the evaluation by the TIA HMM does not significantly influence the performance of a speech recognition system. More empirical results about viability of the TIA HMM will be studied again later when we discuss an optimal state sequence of a speech utterance. Ultimately, of course, recognition performance obtained from a model matters, not absolute likelihood scores.

Simply speaking, finding a HMM producing maximum $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$ is to find a best scoring HMM in light of $P(O_t)$ over $[1, T]$ on the average. Inherently, this implies that if there is, on average, a higher matching rate associated with each individual O_t of a testing utterance to each individual O_t of a training utterance, it is much more probable that the correct utterance is recognized than if there are fewer matching cases. This method is similar to the "perplexity" used in a language model which roughly means the average number of branches at any decision point so that its degree implies the difficulty or uncertainty in each word [4, 33].

In another sense, the likelihood evaluation by the TIA HMM can be regarded as a suboptimal method when considered against the ML criterion, contrary to the conventional F-B HMM.

3.2.2 Experimental Comparisons of Likelihood Between Model Types

In this section, we will empirically show the usefulness of the TIA HMM in speech recognition. The spoken digit recognition problem will be the focus of the experiments. Digit recognition has important applications in on-line banking, credit card inquiry, and automatic dialing.

Fifteen isolated-word utterances of each ten digits "zero" through "nine" were downloaded from ftp://archive.egr.msu.edu/pub/jojo/DPHTEXT. They were collected in a quiet room at the author's lab, recorded on TDK type II using a TEAC W-450R cassette deck with Dolby C noise reduction. Prior to sampling, the data were filtered using an active bandpass, fourth-order Butterworth filter with a low-pass cutoff frequency of 4.7 kHz and a highpass cutoff of 75 Hz. These were uttered by an American adult male and sampled at 10 kHz. A MetraByte DAS16F 12-bit analog to digital conversion board was used to sample the data. Each speech file consists of integer samples covering the range ± 2048. Tenth order cepstral data [c(1) to c(10)] were generated as a feature vector sets from these utterances using 256 points Hamming windows and an FFT algorithm.

These generated cepstral feature vectors were used to construct a codebook of 128 symbols. This codebook was used to quantize the speech sample utterances for training and testing.

The objective of this discussion is to compare the recognition result of two different likelihood measures from the F-B HMM and TIA HMM, and verify that the TIA HMM works properly without much degradation in the speech recognition performance. Since the definitions of likelihood measures from both models are fundamentally different, it is not meaningful to directly compare the likelihood quantities. Instead, we will compute global performance measures from the model types.

For this simulation, a discrete HMM is used as a word model. Due to the available data set, a speaker-dependent model will be tested.

To show the usefulness of the time-invariant state-space HMM empirically, one of the formal tests used in pattern recognition studies is conducted [9].

- In each digit, randomly chosen fourteen isolated words among fifteen utterance are used as a training set for the HMM and then the unselected utterance is used as a testing utterance. Next, the testing utterance is included in the training data set and the other utterance which was chosen as a training utterance previously is assigned as a testing utterance. This procedure is repeated until each utterance is used as a testing utterance once. This procedure is called the *leave-one-out*, or *deleted test* [9].

The experimental results appear in Tables 3.2- 3.3. Each table shows the likelihood result from the F-B HMM and the TIA HMM respectively. For simplicity, only one set of results corresponding to the first of 15 testing utterances is shown here. The results for the other utterances are are similar to those in the tables. $\mathcal{M}_i$ denotes the HMM for the digit i. Five-state Bakis HMMs are used to allow only one skip in any forward transitions.

To avoid numerical underflow caused by the multiplication of many numbers between zero and one, the logarithm with base 10 is taken to the the quantity of $\prod_{t=1}^{T} P(O_t)$. Therefore, the index of the recognized HMM in a given trial is

$$\begin{aligned} & i^* = \arg\max_i \left(\prod_{t=1}^{T} P(O_t \mid \mathcal{M}_i)\right) \\ \Leftrightarrow \quad & i^* = \arg\max_i \left(\log \prod_{t=1}^{T} P(O_t \mid \mathcal{M}_i)\right) \\ \Leftrightarrow \quad & i^* = \arg\max_i \left(\sum_{t=1}^{T} \log P(O_t \mid \mathcal{M}_i)\right) \\ \Leftrightarrow \quad & i^* = \arg\min_i \left(-\sum_{t=1}^{T} \log P(O_t \mid \mathcal{M}_i)\right). \end{aligned} \tag{3.25}$$

In every experiment, the digits are recognized correctly. As well as this recognition result, two additional observations are made.

- Generally, the F-B HMM produces larger likelihood than the TIA HMM. Roughly speaking, this is due to the "extra" probabilistic terms a_{ji}s multiplied to $b_j(O_t)$ as (2.103) in the TIA HMM. Since those a_{ji}s take the values between zero and one, the likelihood decreases as such terms and the cross terms produced by (2.69) increase.

- Even if the recognition performance of two types of model may be the same, there are differences of likelihood. Tables 3.4 and 3.5 are the statistics of Tables 3.2 and 3.3 respectively. They show that the F-B HMM is more advantageous than the TIA HMM from the recognition point of view. This is because not only the average likelihood difference between a correct digit and incorrect digits of the F-B HMM is larger than that of the TIA HMM, but also the variance of incorrect digits of the former is smaller than that of the latter. Therefore, the speech recognition system with the F-B HMM is robust than the TIA HMM. A technique to make TIA HMM robust will be discussed later.

A more fundamental question concerning the speech recognition problem in light of the likelihoods from the F-B HMM and the TIA HMM is the following: For $i \in [1, M]$, if

$$P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}_i) \geq P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}_j) \tag{3.26}$$

holds for all $j \neq i$, is

$$\prod_{t=1}^{T} P(O_t \mid \boldsymbol{\mathcal{M}}_i) \geq \prod_{t=1}^{T} P(O_t \mid \boldsymbol{\mathcal{M}}_j) \tag{3.27}$$

always true for any i? Where M is the number of HMMs which is equivalent to the number of words to be compared. If (3.27) holds for any i, the TIA HMM can be used with equal effectiveness in recognizing strings.

Testing Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	82.1	508.4	572.2	479.0	223.0	529.3	321.4	512.5	182.3	580.0
two	550.7	83.0	779.0	684.1	642.0	634.1	611.6	778.9	672.6	403.6
three	791.6	732.9	109.8	723.6	710.4	731.7	737.6	367.5	753.7	544.3
four	601.8	826.3	670.7	82.9	599.4	753.6	835.3	781.4	839.9	596.2
five	721.6	760.3	601.2	431.2	57.2	706.8	556.5	747.6	680.1	482.0
six	891.1	737.2	850.5	906.3	662.9	140.0	457.0	691.8	873.1	850.2
seven	477.3	676.5	587.9	700.7	492.3	520.4	121.1	708.6	482.2	753.2
eight	575.7	515.0	327.0	608.9	530.3	508.3	565.1	72.2	558.2	525.5
nine	220.6	564.5	652.4	573.3	281.8	610.3	383.0	584.5	60.4	728.2
zero	1053.7	827.9	922.1	840.5	933.4	963.8	944.7	974.2	1027.5	127.3

Table 3.2: Likelihood from the F-B HMM in leave-one-out-test.

Testing Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	112.0	477.4	510.9	490.0	258.9	543.0	340.1	532.2	194.2	508.5
two	630.9	96.2	772.7	707.9	567.9	650.8	628.0	773.7	688.0	434.5
three	793.5	711.5	119.8	726.8	717.1	714.3	746.7	412.1	753.3	427.8
four	630.5	818.7	697.8	103.8	606.1	772.3	840.2	818.5	835.7	481.7
five	724.5	723.1	663.5	394.3	85.9	651.6	581.6	719.2	694.8	506.3
six	894.9	737.5	872.3	897.1	663.3	165.3	338.4	656.6	863.6	854.3
seven	484.3	644.7	646.8	681.0	420.1	481.0	145.1	693.9	469.0	619.4
eight	574.9	516.0	303.9	598.9	516.9	508.9	530.5	95.9	565.5	537.4
nine	179.5	498.2	660.9	623.6	256.8	640.3	411.1	610.4	82.0	651.3
zero	1056.0	840.4	834.1	790.0	876.9	971.9	954.0	960.6	1032.5	172.4

Table 3.3: Likelihood from the TIA HMM based on leave-one-out-test.

Testing Data	Likelihood of correct digit	Likelihood mean of incorrect digits	Likelihood difference of correct digit & incorrect digits	Standard deviation of incorrect digits
one	82.1	434.2	352.1	151.5
two	83.0	639.6	556.6	115.2
three	109.8	677.0	567.2	134.8
four	82.9	722.7	639.8	106.1
five	57.2	631.9	574.7	120.0
six	140.0	768.9	628.9	147.2
seven	121.1	599.9	478.8	110.8
eight	72.2	523.7	451.5	80.5
nine	60.4	510.9	450.5	174.0
zero	127.3	943.0	815.7	75.0
average	93.6	645.2	551.6	121.5

Table 3.4: Statistical Properties of the likelihood results from the F-B HMM.

Testing Data	Likelihood of correct digit	Likelihood mean of incorrect digits	Likelihood difference of correct digit & incorrect digits	Standard deviation of incorrect digits
one	112.0	428.3	316.3	129.7
two	96.2	650.4	554.2	105.5
three	119.8	667.0	547.2	142.4
four	103.8	722.3	618.5	126.6
five	85.9	628.7	542.8	114.3
six	165.3	753.1	587.8	182.7
seven	145.1	571.1	426.0	105.7
eight	95.9	516.9	421.0	85.5
nine	82.0	503.5	421.5	182.2
zero	172.4	924.0	751.6	92.8
average	117.8	636.5	518.7	126.7

Table 3.5: Statistical Properties of the likelihood results from the TIA HMM.

Unfortunately, (3.27) does not hold in general. The relationship is very data-dependent. Therefore, a more practical question is stated as follows: For $i \in [1, M]$, if

$$P(O_1, O_2, \ldots, O_T \mid \mathcal{M}_i) \gg P(O_1, O_2, \ldots, O_T \mid \mathcal{M}_j) \tag{3.28}$$

holds for all $j \neq i$, is

$$\prod_{t=1}^{T} P(O_t \mid \mathcal{M}_i) > \prod_{t=1}^{T} P(O_t \mid \mathcal{M}_j) \tag{3.29}$$

always true for any i? However, it is not easy to analytically specify how much larger the left side of (3.28) needs to be than the right side of (3.28) does. We can only say that in the case of the previous example, the likelihood of the correct word is approximately three times greater than that of the others. More intensive study of the likelihood relationship between the F-B HMM and the TIA HMM in conjunction with the performance of speech recognition is left for future research.

3.2.3 State Probability Distribution Vector in the TIA HMM

In this section, we will investigate $\boldsymbol{x}(t)$ of (2.67) to assess the effect of $\boldsymbol{A}$ in determining state sequences of an utterance in conjunction with the observation symbols.

Consider a Bakis-type TIA HMM. Then, a state-transition equation is composed of $\boldsymbol{A}$ and state probability distribution vector $\boldsymbol{x}(t)$. However, since the state-transition part in the TIA HMM is only composed of $\boldsymbol{A}$ in contrast to $\boldsymbol{A}$ and $\boldsymbol{\Delta}(t)$ in the F-B HMM, a Bakis condition does not have a direct influence on constituting the possible state transitions for an observation string. Instead, $\boldsymbol{A}$ affects $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t)$ indirectly controls state transitions.

To observe the effect of $\boldsymbol{A}$, let us compare two cases, a Bakis ($\boldsymbol{A}_B$) and ergodic ($\boldsymbol{A}_e$) constraints, for state-transition configuration with each other. Tables 3.6 shows

the state probability distribution vectors $\boldsymbol{x}(t)$ for a few specific times when

$$\boldsymbol{A}_B = \begin{pmatrix} 0.9 & 0 \\ 0.1 & 1 \end{pmatrix}, \quad \boldsymbol{A}_e = \begin{pmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{pmatrix}, \tag{3.30}$$

with $\boldsymbol{x}_B(1) = \boldsymbol{x}_e(1) = (1,0)'$.

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=40$	$t=50$	$t=60$	$t=70$
$\boldsymbol{x}_B(t)$	1.000	0.900	0.810	0.729	0.656	0.016	0.005	0.002	0.000
	0	0.100	1.190	0.271	0.343	0.983	0.994	0.998	0.999
$\boldsymbol{x}_e(t)$	1.000	0.900	0.830	0.781	0.746	0.666	0.666	0.666	0.666
	0	0.100	0.170	0.219	0.253	0.333	0.333	0.333	0.333

Table 3.6: State probability distribution vectors under Bakis, $x(t)_B$, and ergodic, $x(t)_e$, constraints.

The effect of $\boldsymbol{A}$ on $\boldsymbol{x}(t)$ is not apparent over short intervals. It follows that the likelihood differences arising from a Bakis and ergodic constraints are not evident over short times. Additionally, it is not possible to distinguish the topology of the HMM from a record of $\boldsymbol{x}(t)$. However, over a sufficient duration of time, we see that the difference in two state probability distribution vectors becomes distinguishable.

As explained in Section 3.2.1, the effect of $\boldsymbol{A}$ in the TIA HMM is indirect and "global" in a sense to form "possible" state paths in an utterance, contrary to $\boldsymbol{\Delta}$ which locally affects state paths.

3.2.4 Comparison of $\boldsymbol{x}(t)$ and $\boldsymbol{\gamma}(t)$

In this section, we will discuss similarities of one state variable of the F-B HMM and the other of the TIA HMM. Particularly, we are interested in the similarity of state probability distribution vector $\boldsymbol{x}(t)$ of the TIA HMM and $\boldsymbol{\gamma}(O_t)$ of the F-B HMM.

Previously, we have

$$x_i(t) = P(q_t = i \mid \boldsymbol{\mathcal{M}}) \tag{3.31}$$

$$\gamma_i(t) = P(q_t = i \mid \boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{M}}), \tag{3.32}$$

where $\boldsymbol{\mathcal{O}} = \{O_1, O_2, \ldots, O_T\}$ and $\boldsymbol{\mathcal{M}} = \{N, M, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. Note that $\gamma_i(t)$ is the *a posteriori* probability of a state based on $\boldsymbol{\mathcal{O}}$. On the other hand, $x_i(t)$ is a state probability distribution without $\boldsymbol{\mathcal{O}}$ although both state variables provide information about the probability being in state i at time t.

Reconsider the previous recognition experiments. For analysis purposes, consider the case for word "four." Suppose that $\boldsymbol{\mathcal{M}}$ is computed from fifteen training utterances using the F-B HMM and $\boldsymbol{A}$ is

$$\boldsymbol{A} = \begin{pmatrix} 0.9625 & 0 & 0 & 0 & 0 \\ 0.0375 & 0.8835 & 0 & 0 & 0 \\ 0 & 0.1165 & 0.9704 & 0 & 0 \\ 0 & 0 & 0.0277 & 0.8652 & 0 \\ 0 & 0 & 0.0020 & 0.1348 & 1.0000 \end{pmatrix}. \tag{3.33}$$

The corresponding state probability distribution $x_i(t)$ along t for $i = 1, \ldots, 5$ appears in Fig. 3.1. Also, with $\boldsymbol{\mathcal{M}}$, $\gamma_i(t)$ $i = 1, \ldots, 5$ can be computed for each utterance $\boldsymbol{\mathcal{O}}$ of training data set. It is interesting when all $\gamma_i(t), i = 1, \ldots, 5$ of the training data set is combined and the average of the $\gamma_i(t)$ is computed along t. Here since the length of each training utterance may be different, it is not possible to get complete alignment of the training data set along t. Instead, only the time duration commonly occupied by all training data set is considered.

The average of $\gamma_i(t)$ for the entire training utterance is shown in Fig. 3.2. Comparing results, the transition pattern of $x_i(t)$ is seen to be similar to that of $\gamma_i(t)$ for

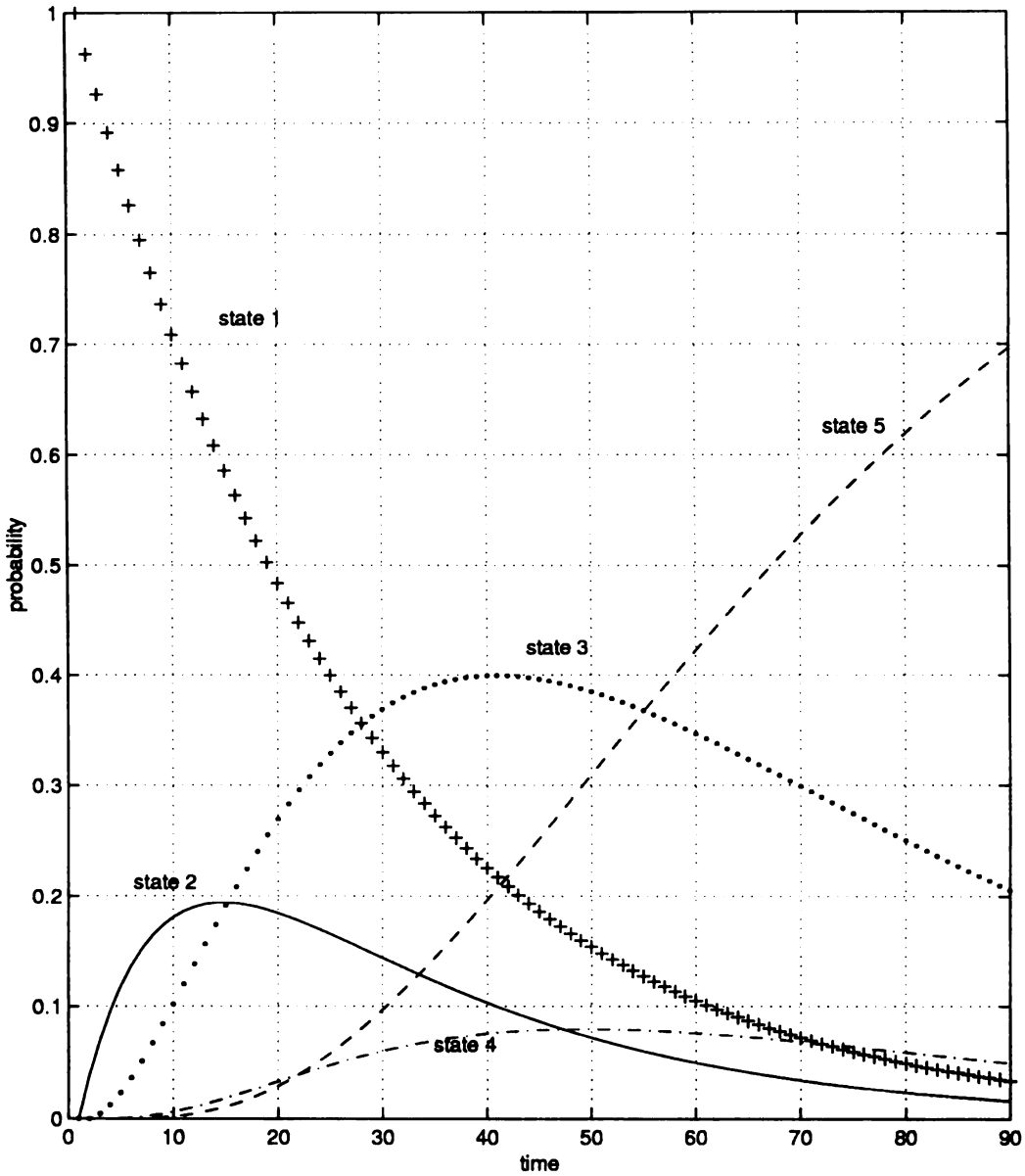

Figure 3.1: State probability distribution after training digit "four."

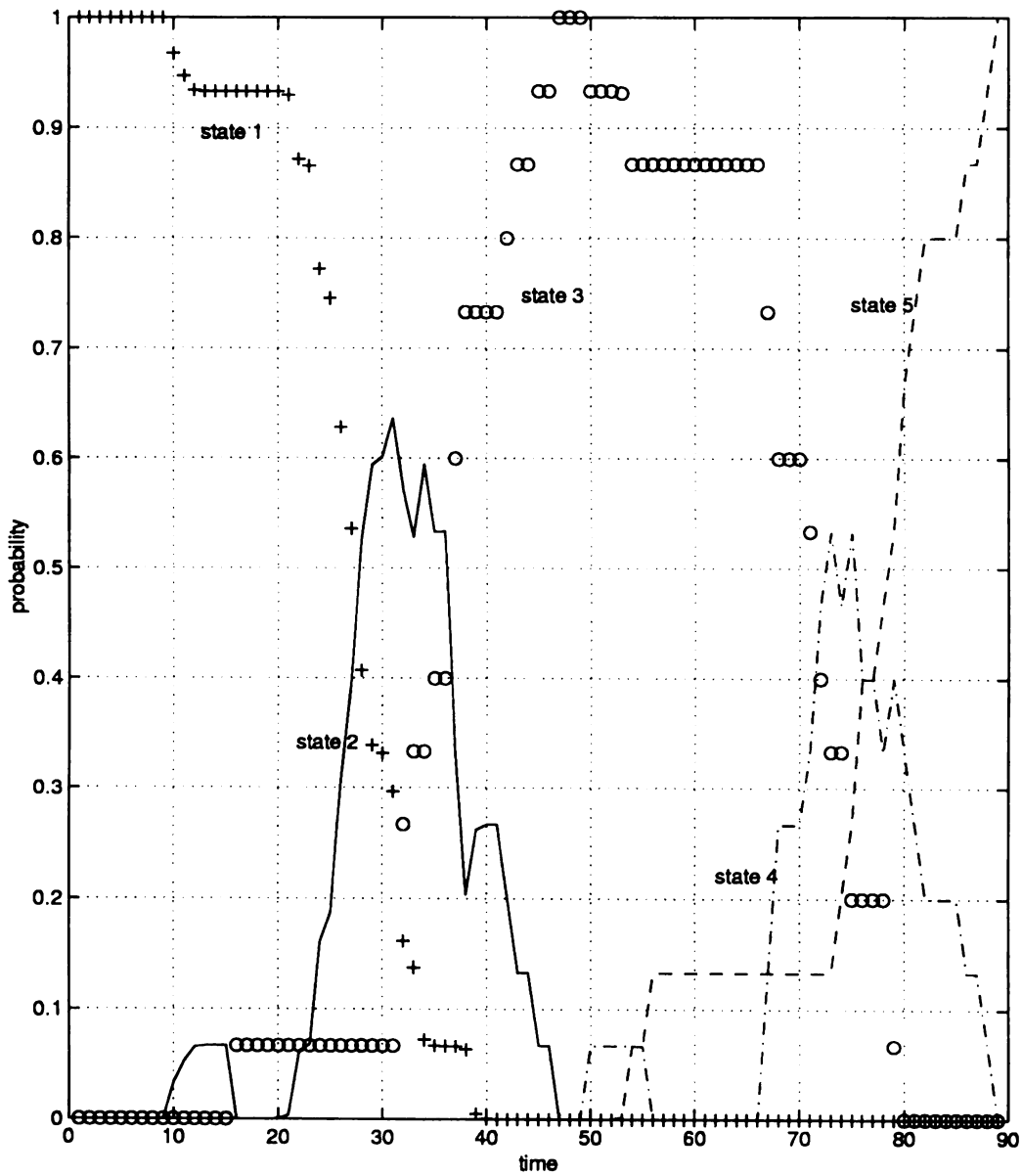

Figure 3.2: The average of $\gamma_i(t), i = 1, \ldots, 5$ from the entire training utterances of word "four."

each i even though the actual probabilities are different. For the other digits other than word "four," we see the same phenomenon.

It is not easy to say how these two statistical variables from different models is related since $x_i(t)$ is a infinite sequence if t is not constrained and $\gamma_i(t)$ is a finite sequence according to the size of training data. Roughly speaking, however, the average of $\gamma_i(t)$ for the training data set amounts to the state probability distribution $x_i(t)$. This phenomenon is related to counting process when $\boldsymbol{A}, \boldsymbol{B}$ are computed.

3.2.5 Experimental Results on the Effects of $\boldsymbol{A}$

We have shown theoretically that a strongly diagonal $\boldsymbol{A}$ does not make significant contribution to the likelihood scores in the TIA HMM. Here, we will show this experimentally with some examples. Along with this experiment, we will discuss the possible way of reducing the computational loads required in the HMM training using the characteristics of $\boldsymbol{A}$.

To observe the effect of $\boldsymbol{A}$ of the TIA HMM, first let us update $\boldsymbol{B}$ only in the HMM training in the five-state Bakis HMMs while allowing only one skip in any forward transitions. In other words, after $\boldsymbol{A}$ is assigned initially, the training procedure estimates $\boldsymbol{B}$ only and does not change $\boldsymbol{A}$. Then, compare the likelihoods. Let

$$\boldsymbol{A}_{B_1} = \begin{pmatrix} 0.99 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.01 & 0.99 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.01 & 0.90 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.05 & 0.99 & 0.00 \\ 0.00 & 0.00 & 0.05 & 0.01 & 1.00 \end{pmatrix}, \tag{3.34}$$

$$\boldsymbol{A}_{B_2} = \begin{pmatrix} 0.70 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.20 & 0.80 & 0.00 & 0.00 & 0.00 \\ 0.10 & 0.10 & 0.60 & 0.00 & 0.00 \\ 0.00 & 0.10 & 0.30 & 0.80 & 0.00 \\ 0.00 & 0.00 & 0.10 & 0.20 & 1.00 \end{pmatrix} \tag{3.35}$$

be the two preset state-transition matrices, for example. $\boldsymbol{A}_{B_1}$ is more diagonally dominant than $\boldsymbol{A}_{B_2}$. $\boldsymbol{A}_i$ is the state-transition matrix for digit i from the F-B HMM training. Table 3.7 is the sum of likelihoods of fifteen training utterances for each digit. Note that since we take the negative log to the likelihood for numerical purpose, the ML actually amounts to the minimum likelihood in the table.

	$\boldsymbol{A}_i$	$\boldsymbol{A}_{B_1}$	$\boldsymbol{A}_{B_2}$
$\mathcal{M}_1$	1008.4	1019.5	1053.3
$\mathcal{M}_2$	1111.3	1053.1	1074.0
$\mathcal{M}_3$	1154.5	1176.0	1240.6
$\mathcal{M}_4$	1182.2	1120.5	1227.3
$\mathcal{M}_5$	1030.7	1072.5	1146.7
$\mathcal{M}_6$	2111.8	1996.4	2123.1
$\mathcal{M}_7$	1725.6	1632.3	1719.8
$\mathcal{M}_8$	1028.3	991.3	1006.9
$\mathcal{M}_9$	1073.2	899.9	991.8
$\mathcal{M}_0$	1908.0	1689.5	1800.7

Table 3.7: Sum of likelihoods of fifteen training utterances for each digit associated with three different state-transition matrices in the F-B HMM.

From the table, we see that the likelihoods from the usual F-B HMM which requires both $\boldsymbol{A}$ and $\boldsymbol{B}$ training can be frequently less than those of the models whose $\boldsymbol{A}$ is arbitrarily set and only $\boldsymbol{B}$ is updated. Other than the problem of local minimum of the HMM training in the optimization criterion, we see that the training $\boldsymbol{A}$ is not much crucial in certain cases such as having diagonally dominant $\boldsymbol{A}$ in the HMM.

Next, to examine the recognition results for different state-transition matrices, let the *resubstitution-test* be performed even though such a test is not practical in speech recognition system. In the resubstitution-test, the training utterance is used for a testing utterance. In this simulation, however, there is no difference between a resubstitution-test and leave-one-out-test because we are looking for the effect of the topology of $\boldsymbol{A}$ in the F-B HMM and TIA HMM. We can reach the the same conclusions with a leave-one-out-test. Also, the results from the resubstitution-test will be useful when we discuss the topic about finding an optimal state sequence in a speech utterance in Chapter 4.

The recognition results are in Table 3.8 through Table 3.13. Table 3.8 shows the likelihoods for each digit from the F-B HMM computation when one randomly chosen testing utterance among fifteen is evaluated by the HMMs. Table 3.9 shows the likelihoods for each digit from the TIA HMM computation when the same testing utterance in the case of F-B HMM is evaluated by the HMMs. Table 3.10 and Table 3.11 are the likelihoods for $\boldsymbol{A}_{B_1}$ for the F-B HMM and the TIA HMM respectively. On the other hand, Table 3.12 and Table 3.13 are for $\boldsymbol{A}_{B_2}$.

Comparing Table 3.10-Table 3.13 with Table 3.8-Table 3.9, yields the following observations:

- The digit recognition performance with $\boldsymbol{A}_{B_1}$ and $\boldsymbol{A}_{B_2}$ matches the performance with $\boldsymbol{A}_1$ and $\boldsymbol{B}$ in the F-B HMM. In addition, the more diagonally dominant $\boldsymbol{A}$ is, the better the recognition performance.

- In case of the TIA HMM, we have the same conclusion that the more diagonally dominant $\boldsymbol{A}$ is, the better the recognition performance. However, the recognition performance is more sensitive to the values of state-transition matrix than that of the F-B HMM.

- In an extreme case such as $a_{jj} = \frac{1}{N}$ for all $j \in [1, N]$, the recognition per-

Test data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	63.7	481.0	569.0	553.8	218.1	532.4	320.4	534.5	173.7	567.1
two	476.2	84.3	669.2	570.5	643.7	638.8	489.7	635.0	724.1	456.8
three	800.7	758.5	75.3	728.7	682.7	697.8	741.8	544.3	747.3	493.6
four	593.2	680.2	711.9	70.7	612.0	757.4	843.7	823.2	868.0	559.5
five	727.4	748.7	622.7	442.9	53.1	637.5	592.1	696.5	694.7	498.7
six	857.2	681.5	756.8	820.3	633.0	122.7	302.2	536.1	711.7	827.9
seven	473.3	654.9	517.1	655.2	365.9	526.8	96.1	685.0	382.7	670.3
eight	562.7	536.1	424.4	574.7	530.2	447.8	524.9	66.8	532.9	551.7
nine	194.7	551.4	629.4	677.9	265.8	621.8	366.9	580.0	63.5	677.6
zero	1054.1	833.4	898.6	839.9	893.2	976.0	940.0	986.3	366.0	129.9

Table 3.8: Likelihood $P(\mathcal{O} \mid \mathcal{M})$ using $\boldsymbol{A}_i$ and $\boldsymbol{B}$ for each digit i in a resubstitution test.

Test Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	93.8	457.4	511.6	555.9	251.1	543.2	341.8	526.6	195.8	610.7
two	549.9	106.6	627.2	619.1	557.8	654.5	503.3	626.7	729.2	424.6
three	809.8	725.7	111.8	721.3	678.4	709.3	750.3	547.2	729.8	475.9
four	611.5	689.9	710.9	103.5	612.8	796.5	847.8	838.6	872.7	490.0
five	729.3	714.3	650.9	413.6	97.2	642.6	610.5	712.5	702.1	531.7
six	860.9	705.3	771.5	806.9	639.1	145.2	297.4	553.5	773.6	815.2
seven	413.8	638.2	575.7	699.7	391.0	539.3	122.7	700.8	409.4	683.2
eight	568.2	479.2	360.7	548.9	520.3	449.7	510.2	84.8	543.8	559.1
nine	139.3	487.7	634.9	692.3	231.3	647.5	395.6	588.1	95.8	694.0
zero	1056.0	842.8	818.3	789.2	862.0	981.5	952.5	977.3	1041.7	160.9

Table 3.9: Likelihood $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$ using $\boldsymbol{A}_i$ and $\boldsymbol{B}$ for each digit i in a resubstitution test.

Test Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	65.2	532.9	569.5	550.1	214.1	530.9	322.0	517.0	170.6	577.8
two	471.2	78.7	675.8	580.3	633.6	680.3	481.9	636.1	724.3	443.5
three	801.6	755.1	76.5	733.8	681.2	755.1	745.1	557.1	755.4	483.5
four	587.6	672.2	711.2	70.6	604.1	759.4	849.6	826.1	869.1	572.3
five	729.2	743.1	622.2	429.0	59.7	692.1	585.7	757.9	690.7	491.8
six	858.1	697.0	755.9	807.0	639.9	115.5	307.4	599.3	715.6	850.2
seven	473.9	665.3	516.6	666.3	376.8	553.0	88.8	740.9	461.6	741.1
eight	563.5	536.6	424.6	573.2	537.8	464.4	519.4	72.4	535.8	529.0
nine	173.5	604.9	628.9	673.3	273.0	621.2	371.2	566.2	52.7	673.1
zero	1054.9	828.2	897.5	823.6	950.3	975.0	945.0	989.2	1040.4	114.0

Table 3.10: Likelihood $P(\mathcal{O} \mid \mathcal{M})$ using $\boldsymbol{A}_{B_1}$ and $\boldsymbol{B}_{\boldsymbol{A}_{B_1}}$ for each digit in a resubstitution test.

Test Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	175.8	482.5	514.0	567.7	300.4	570.9	385.2	557.4	208.5	510.4
two	470.7	166.6	615.7	595.5	556.2	682.3	546.7	644.1	724.0	453.7
three	827.4	740.9	161.0	725.8	689.3	743.8	760.5	547.2	741.0	479.1
four	607.2	725.2	694.4	134.2	610.0	802.4	853.7	852.7	876.0	533.5
five	741.9	704.1	619.9	436.9	125.8	665.6	608.9	740.5	669.6	550.5
six	877.0	660.9	743.2	782.3	664.8	197.1	220.9	582.5	724.9	747.4
seven	451.0	666.7	527.6	685.0	432.6	567.3	177.0	722.6	402.6	576.8
eight	583.4	445.1	366.3	537.2	525.9	461.6	509.2	120.6	557.5	478.0
nine	223.0	513.6	634.7	712.0	296.8	673.9	437.8	628.9	99.2	599.2
zero	1056.0	853.9	819.9	787.1	853.2	995.6	946.9	991.6	1044.8	208.9

Table 3.11: Likelihood $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$ using $\boldsymbol{A}_{B_1}$ and $\boldsymbol{B}_{\boldsymbol{A}_{B_1}}$ for each digit in a resubstitution test.

Test Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	66.2	470.1	566.6	548.1	212.3	536.6	320.9	534.0	179.0	583.5
two	540.9	81.7	681.8	586.2	645.5	639.8	486.6	626.7	726.4	447.6
three	798.3	760.5	81.4	733.8	676.2	697.9	734.9	542.1	716.2	543.5
four	592.6	672.1	706.4	77.7	617.3	748.6	847.6	821.5	867.3	568.2
five	727.1	755.7	617.2	433.0	61.8	643.1	595.7	694.8	690.8	493.3
six	854.8	687.9	804.7	817.0	636.1	123.7	294.1	528.0	720.4	849.4
seven	476.7	650.4	514.3	673.8	376.8	530.4	93.5	677.4	369.6	754.1
eight	560.2	530.2	424.3	577.5	528.3	449.9	525.7	68.7	535.8	558.1
nine	178.9	535.2	627.8	670.0	271.5	628.0	366.5	575.8	59.8	671.4
zero	1051.6	831.6	911.4	830.3	953.7	978.7	952.0	989.3	1038.4	121.2

Table 3.12: Likelihood $P(\mathcal{O} \mid \mathcal{M})$ using $\boldsymbol{A}_{B_2}$ and $\boldsymbol{B}_{\boldsymbol{A}_{B_2}}$ for each digit in a resubstitution test.

Test Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	135.5	470.3	569.8	554.1	306.0	641.6	358.0	536.3	255.2	626.5
two	710.4	206.4	707.7	774.0	680.0	677.2	710.2	613.9	746.7	475.5
three	792.9	775.5	197.6	749.4	676.2	730.1	744.8	725.9	712.6	728.4
four	745.0	672.8	773.8	320.9	705.6	761.5	860.6	828.0	871.8	566.0
five	731.5	825.8	773.0	607.9	441.0	822.4	811.9	702.0	798.5	651.6
six	859.2	919.5	878.3	943.4	739.3	260.8	662.5	536.9	850.5	906.4
seven	491.6	655.6	769.7	844.1	554.8	619.3	381.2	687.4	590.4	759.4
eight	564.6	583.7	417.8	640.4	532.7	481.7	568.0	141.5	540.9	623.0
nine	236.5	540.5	684.0	674.9	406.9	769.5	486.2	584.9	292.3	734.9
zero	992.0	892.5	945.5	989.5	1041.1	999.7	1017.7	980.8	1046.0	611.0

Table 3.13: Likelihood $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$ using $\boldsymbol{A}_{B_2}$ and $\boldsymbol{B}_{\boldsymbol{A}_{B_2}}$ for each digit in a resubstitution test.

formance becomes degraded. This is because of the more ambiguity caused by "equally likely" state transition. Contrary to this argument, with $\boldsymbol{A}$ which is close to a diagonalized matrix, the performance becomes enhanced because of lessened ambiguity about state occupancy at a given time. In information theory, such uncertainty is measured in terms of entropy [5].

For the leave-one-out tests, the conclusions above are the same.

Associated with these experimental results, we see that the required time and resources to train the HMMs become lessened with a preset state-transition matrix $\boldsymbol{A}_{B_1}$ due to the unnecessity of training a state-transition matrix. Therefore, it is possible to reduce the computational loads required in the training.

Followed by this assertion, the arising problem could be "how close does $\boldsymbol{A}$ need to be a diagonalized matrix to obtain a satisfactory recognition performance?". This is left for future research.

3.3 Reconciliation of the TIA HMM

In this section, we are going to discuss a few evolving techniques that reconcile the TIA HMM to the conventional F-B HMM.

3.3.1 Feedback Control

From Table 3.10 through 3.13, we see that the less each $x_i(t)$ is overlapped, the better the performance. The implies that in the TIA HMM, the closer one of states is probability one at each t, the better the performance. Simply speaking, we want a TIA HMM close enough to a certain "unknown" desirable system so that the states are separable from each other as much as possible for every t. Such separation can decrease the adverse effects of illegal paths.

However, the exponentially decaying characteristic by the Markovian assumption in the HMM basically does not allow such "separation." One attempt to compensate for extra probabilities, as well as to decrease illegal paths effects, is to use state-variable feedback to get a desired state responses. State feedback technique is to relocate the eigenvalues of a system to get a desired system response [94].

If a given linear time-invariant system realization is state controllable, any desired characteristic polynomial can be obtained by state-variable feedback. In our problem, however, neither do we have specific desired eigenvalues, not do we know exactly which eigenvalues will be optimal in a sense that the recognition performance as well as its robustness of the TIA HMM is comparable to those of the F-B HMM. Provided that such desirable poles are known, we have

$$\boldsymbol{x}(t+1) \;=\; \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{u}(t)\delta(t) + \boldsymbol{W}(t), \tag{3.36}$$

from the TIA HMM. Where $\boldsymbol{W}$ is the state feedback input which regulates the state probability so that $\boldsymbol{x}(t+1)$ can reach the desired values for each t by

$$\boldsymbol{W}(t) = \boldsymbol{F}\boldsymbol{x}(t). \tag{3.37}$$

Here we do not have a specific control input except an initial time $t = 0$ in the HMM. This is in contrast to the usual state-space control problem. Therefore, to accommodate the time-varying nature of a speech signal and to avoid the exponential decaying state probabilities of the Markovian model, we need a time-varying feedback control such as

$$\boldsymbol{W}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t). \tag{3.38}$$

Unfortunately, this attempt does not allow us to have stationary diagonalized state-

transition matrix $[\boldsymbol{A} + \boldsymbol{F}(t)]$ for all t. Therefore, we have the same problem which we face in the F-B HMM. The motivation for using the TIA HMM in speech recognition lies on its diagonalization. Therefore, the principal advantage of using the diagonalization of the state equation does not exist in this approach.

3.3.2 Stochastic Modeling of Temporal Information in the TIA HMM

To make the TIA HMM robust, consider a model which includes additional temporal information between neighboring symbols of an utterance.

The assumption that the observations generated by the HMM's hidden process are only state dependent is, in fact, a limitation of the HMM when applied to a real speech signal. In reality, speech features are correlated. To include additional time-ordering relation between consecutive symbols in a speech utterance, consider one of the techniques proposed by Dai *et al.* [63]. The idea is to include a Markovian relation between symbols instead of just "observation independent."

In Dai's approach, the state-space is the codebook and each symbol in the codebook becomes a state of the Markov process. The revised criterion seeks to find a HMM which produces a ML in the conventional F-B HMM sense in conjunction with the likelihood based on the Markovian relation between symbols as

$$L^{'}(\boldsymbol{\mathcal{O}}) \;=\; P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}}^{'}), \tag{3.39}$$

where

$$P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}}) \;=\; P(O_1, O_2, \ldots, O_T \mid \boldsymbol{\mathcal{M}}), \tag{3.40}$$

and

$$\begin{aligned} P(\mathcal{O} \mid \mathcal{M}') &= P(O_1, O_2, \ldots, O_T \mid \mathcal{M}') \\ &= P(O_T \mid O_1, O_2, \ldots, O_{T-1}, \mathcal{M}') P(O_1, O_2, \ldots, O_{T-1} \mid \mathcal{M}') \\ &= P(O_T \mid O_{T-1}, \mathcal{M}') P(O_1, O_2, \ldots, O_{T-1} \mid \mathcal{M}') \\ &\vdots \\ &= \prod_{t=1}^{T} P(O_t \mid O_{t-1}, \mathcal{M}') \end{aligned} \tag{3.41}$$

with

$$P(O_1 \mid O_0, \mathcal{M}') = P(O_1 \mid \mathcal{M}'). \tag{3.42}$$

Here $\mathcal{M}'$ stands for the set of initial symbol probability and symbol transition matrix.

The same idea can be applied to the TIA HMM, the likelihood $\prod_{t=1}^{T} P(O_t \mid \mathcal{M})$, as

$$\begin{aligned} L'(\mathcal{O} \mid \mathcal{M}, \mathcal{M}') &= (\prod_{t=1}^{T} P(O_t \mid \mathcal{M}))(P(\mathcal{O} \mid \mathcal{M}')) \\ &= (\prod_{t=1}^{T} P(O_t \mid \mathcal{M}))(\prod_{t=1}^{T} P(O_t \mid O_{t-1}, \mathcal{M}')) \\ &= \prod_{t=1}^{T} (P(O_t \mid \mathcal{M}) P(O_t \mid O_{t-1}, \mathcal{M}')). \end{aligned} \tag{3.43}$$

Therefore,

$$\begin{aligned} L(\mathcal{O} \mid \mathcal{M}, \mathcal{M}') &= -\log \prod_{t=1}^{T} (P(O_t \mid \mathcal{M}) P(O_t \mid O_{t-1}, \mathcal{M}')) \\ &= -\sum_{t=1}^{T} \log(P(O_t \mid \mathcal{M}) P(O_t \mid O_{t-1}, \mathcal{M}')) \\ &= -\sum_{t=1}^{T} (\log P(O_t \mid \mathcal{M}) + \log P(O_t \mid O_{t-1}, \mathcal{M}')). \end{aligned} \tag{3.44}$$

The $L(\mathcal{O} \mid \mathcal{M}, \mathcal{M}')$ likelihood results for the ten spoken digits database described in Section 3.2.2 are given Table 3.14. Here ∞ in the table denotes infinite value caused

Testing Data	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_5$	$\mathcal{M}_6$	$\mathcal{M}_7$	$\mathcal{M}_8$	$\mathcal{M}_9$	$\mathcal{M}_0$
one	91.0	∞	∞	∞	∞	∞	∞	∞	∞	∞
two	∞	108.3	∞	∞	∞	∞	∞	∞	∞	∞
three	∞	∞	108.9	∞	∞	∞	∞	∞	∞	∞
four	∞	∞	∞	112.3	∞	∞	∞	∞	∞	∞
five	∞	∞	∞	∞	91.9	∞	∞	∞	∞	∞
six	∞	∞	∞	∞	∞	147.9	∞	∞	∞	∞
seven	∞	∞	∞	∞	∞	∞	116.0	∞	∞	∞
eight	∞	∞	∞	∞	∞	∞	∞	84.0	∞	∞
nine	∞	∞	∞	∞	∞	∞	∞	∞	96.3	∞
ten	∞	∞	∞	∞	∞	∞	∞	∞	∞	165.5

Table 3.14: The likelihoods based on $L(\mathcal{O} \mid \mathcal{M}, \mathcal{M}')$ with the TIA HMM.

by the negative $\log 0$. It is obvious that all the digits are recognized correctly. As well as this correct recognition performance, the difference of likelihood measure (or variance in likelihood measure) between digit i and digit $j \neq i$ evaluated by $\mathcal{M}_i$ becomes large. As a result, the recognition system is robust.

3.4 Discussion

In this chapter, the problem caused by mismatch between two model types, the TIA and F-B HMM, have been focused on in detail. Additionally, practical issues in the use of the TIA HMM have been discussed with the theoretical and empirical evidences of the model.

Regardless of the problem caused by illegal state sequences, with the TIA HMM, we obtain the comparable speech recognition performance to the F-B HMM in some applications such as digit recognition. Such flexibility of controlling recognition rate

and speed and memory makes the TIA HMM useful some applications.

Next, to reconcile the TIA HMM to the F-B HMM, various attempts are taken although both models theoretically cannot be identical. Through those approaches, however, two significant results are reverified. First, albeit the inherent difference of both models, there are some similarities between certain state variables of each model. Next, the relative importance of $\boldsymbol{B}$ over $\boldsymbol{A}$ was reverified. Also, it was found that the diagonally dominant condition on the state-transition matrix in the TIA HMM is an important factor to affect the the performance of speech recognition.

Finally, we introduced a possible technique to render the TIA HMM robust. The technique is to add one more temporal constrain between symbols of an utterance to the existing HMM for HMM evaluation. Although the technique requires additional memory and computations for this new constraint, it increases the robustness of speech recognition.

Chapter 4

Training HMMs so that Hidden Model States Meaningfully Represent Acoustic States

The HMM is a state model and a speech utterance is modeled to be generated in accordance with state transitions. In particular, a meaningful state sequence is significant. Through a meaningful state sequence, for example, we can learn about the structure of the signal model, and obtain the average statistics of the individual states. In addition, the experimental evidence suggests that a state frequently represents one or more identifiable acoustic phenomena [2, 4]. Thus, we can discover the acoustic characteristics of a speech utterance associated with such a meaningful state sequence.

Finding a meaningful state sequence of a speech signal in the HMM is often cited as one of three major analytical problems centered on the HMM. The information about an appropriate state sequence is useful to improve the performance of speech recognition system in conjunction with the solutions of two other HMM problems, the evaluation as well as the training. This is because the evaluation and training

through a meaningful state sequence produces better performance and gives simple algorithms for evaluation and training.

Like the evaluation problem of the HMM, for which a solution can be given depending on the likelihood criterion, there are several possible ways to find a meaningful state sequence corresponding to a given observation sequence. Depending on the optimality criterion followed by the definition of " state sequence," the result of possible state sequences corresponding to a speech signal may be different.

The problem of finding a meaningful state sequence involves the attempt to uncover the hidden part of the model. Depending on the application, different criteria can be employed to find an optimal state sequence [56, 57, 59]. Among them, the Viterbi search is a prevalent one and it is based on the probability that the HMM could generate the observation sequence using the best possible state sequence [57, 58],

$$Q^* = \arg\max_{Q} P(\mathcal{O}, Q \mid \mathcal{M}), \tag{4.1}$$

in which Q represents any state sequence of length T.

The specific goal in this chapter is to propose some new techniques for finding a meaningful state sequence associated with a speech utterance. These techniques are used for training HMMs in meaningful ways. Then, the results from each search technique are compared to those of the Viterbi of the HMM.

In this discussion, it is assumed that a speech signal is already encoded with reference to a codebook of 128 unique spectral vectors. Hence, a speech utterance is the sequence of codebook indices represented in the abstract as $\{O_1, O_t, \ldots, O_T\}$. In order to reduce the computational complexities required for recognition and analysis, this research is restricted to the recognition of isolated words based on the discrete HMM.

4.1 Maximum Likelihood Approach to State Sequence Determination

We showed previously the usefulness of the TIA HMM in a spoken digit recognition problem. Also, the TIA HMM is potentially advantageous in a large-vocabulary system for computationally efficiency. As well as evaluating the likelihood, the TIA HMM technique can also be useful to find a meaningful state sequence of a speech signal. One way to enhance the performance of speech recognition is to exploit the state sequence information which is significant to compute more informative parameters for $\mathcal{M}$ during training HMMs. Let us discuss this topic in detail.

4.1.1 Introduction

The Viterbi search technique is used to find an optimal state sequence of a speech signal based on the criterion (4.1) in the F-B HMM. As well as this widely used technique, however, there may potentially exist many other search techniques. Consider one of the possible criteria for finding state sequence under the framework of the HMM as

$$q_t^* = \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}) \quad \text{for } 1 \leq t \leq T \tag{4.2}$$

and

$$Q^* = \{q_1^*, q_2^*, \ldots, q_T^*\}. \tag{4.3}$$

In other words, an optimal state sequence is a sequence of individuals state at each t which is most likely to produce a symbol at t in conjunction with a state distribution probability. Without an explicit imposing constraint on state transitions, this

criterion is simpler than (4.1) which requires a backtracking procedure.

Actually, this criterion is close to the formulation of the TIA HMM. In fact, the state searching procedure of the F-B HMM is to Viterbi what that of the TIA HMM is to (4.3). Apparently, (4.2) and (4.3) can be easily implemented by the TIA HMM since

$$q_t^* = \max_{q_t} P(O_t, q_t \mid \mathcal{M}) \tag{4.4}$$

$$= \max_{q_t} \{P(O_t \mid q_t, \mathcal{M}) P(q_t \mid \mathcal{M})\} \tag{4.5}$$

$$= \max_{1 \le i \le N} \{b_i(O_t) x_i(t)\}. \tag{4.6}$$

A Bakis condition on state-transition matrix is not "explicitly" involved in in constituting the legal state sequence by (4.2) and (4.3). Because of lack of explicit imposing constraint on state transitions, the TIA HMM produce an illegal state sequences during operations.

In Chapter 3, however, we have shown that the recognition performance is comparable to the conventional F-B HMM in spite of illegal state sequences. By the same token, it will be shown here that the criterion (4.2) and (4.3) is also practically efficient in finding a meaningful state sequence.

It will be shown that even though the computed state sequence of a speech utterance by (4.2) and (4.3) is not completely identical to the state sequence by the Viterbi search technique for all $t \in [1, T]$, however, remarkably, the result of this technique is quite close to that of the Viterbi search in a "global sense" for correct word. The global shapes of computed state sequences by the Viterbi search and this new ML in accordance with the TIA HMM are similar to each other. Moreover, it will be discussed that this ML technique is a fast and suboptimal method to obtain a possible state sequence information without backtracking procedure required in the Viterbi search.

4.1.2 Experimental Results

To see the viability of the search criterion (4.2), (4.6) is applied to find a meaningful state sequence of the four different spoken utterances of "one," "two," "four," and "six." The resubstitution and leave-one-out test are performed with these example utterances. Here, it is assumed that $\mathcal{M}_i$ for all $i = 1, \ldots 10$ is computed in advance using the F-B algorithm. Index $i = 1$ represents word "one," and $i = 10$ represent word "zero."

To assess the effectiveness of the TIA HMM at finding a meaningful state sequence for speech signals, let us apply (4.6) to the four example utterances.

Figures 4.1 through 4.8 are computed state sequences of four individual testing word composed of "one," "two," "four," and "six" based on $\mathcal{M}_i$, $i = 1, \ldots, 10$. Figures 4.1 through 4.4 are the resubstitution results and Figs. 4.5 through 4.8 are the leave-one-out test results for the same utterances. Each figure is also composed of ten subplots. For example, in Fig. 4.1, the left top figure is the raw speech waveform of the original spoken digit of word "one." Below this raw waveform are the state search results using the conventional Viterbi algorithm with the F-B HMM formulation based on $\{\boldsymbol{A}_i, \boldsymbol{B}_i\}$, where $i = 1, 2, 4, 7$, respectively. For example, in Fig. 4.1, the Viterbi search result is obtained when the testing word "one" is evaluated by $\{\boldsymbol{A}_1, \boldsymbol{B}_1\}$ which is the trained HMM for the word "one." When that utterance is evaluated by different HMMs other than $\{\boldsymbol{A}_1, \boldsymbol{B}_1\}$, the Viterbi algorithm does not provide reasonable results except the cases of mis-recognition. Therefore, the Viterbi state search results for other words not drawn.

The rest of the figures are state search results based on criteria (4.2) and (4.3) for $\mathcal{M}_i$, $i = 1, \ldots, 10$ ranging from top to bottom in the left and right columns, respectively.

From the figures, the following are observed.

- As expected, the resubstitution method produces better result than the leave-

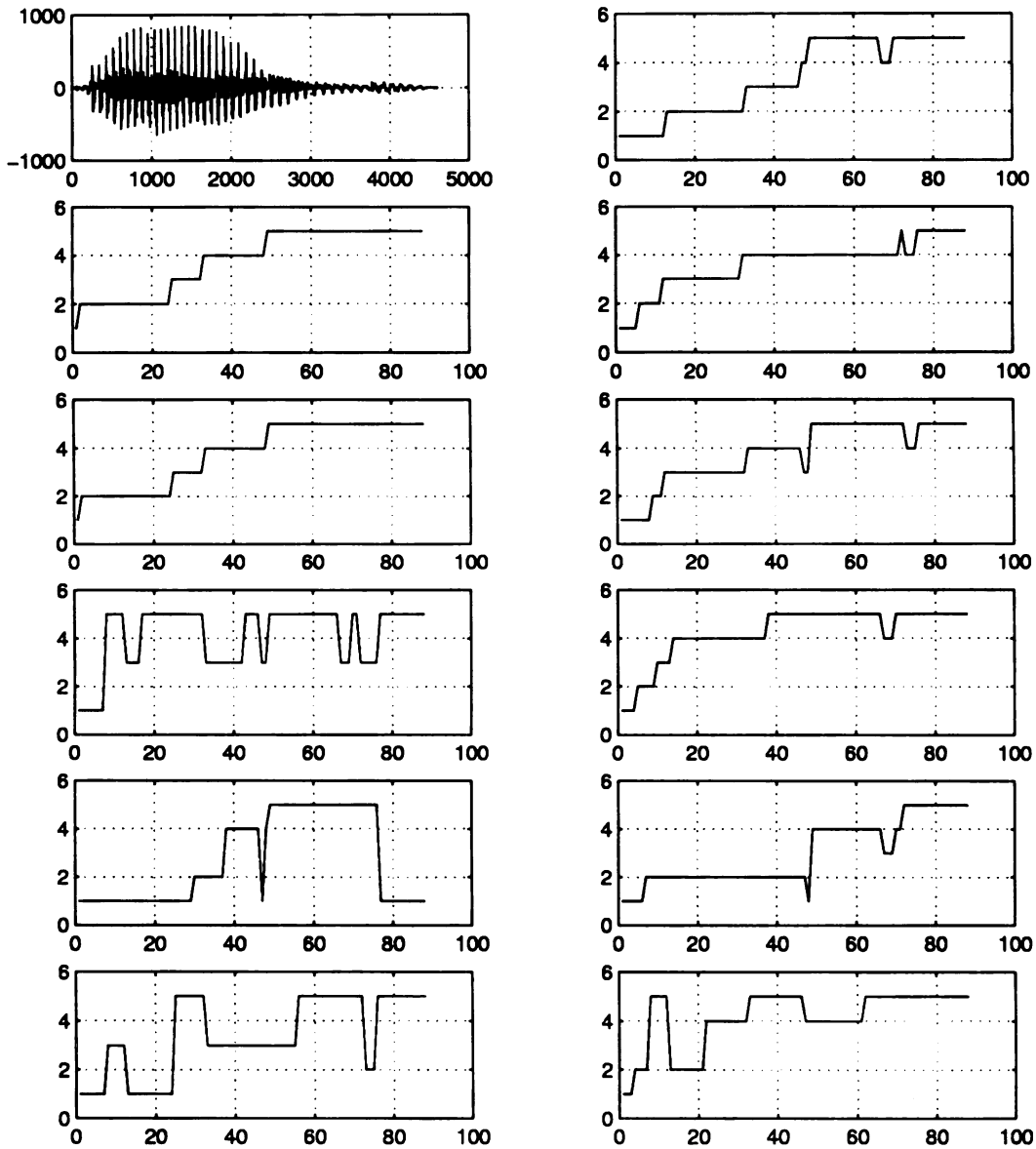

Figure 4.1: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i), \quad i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "one." Note that each graph represents a different "i" except top two figures in the left column. The tests employ resubstitution.

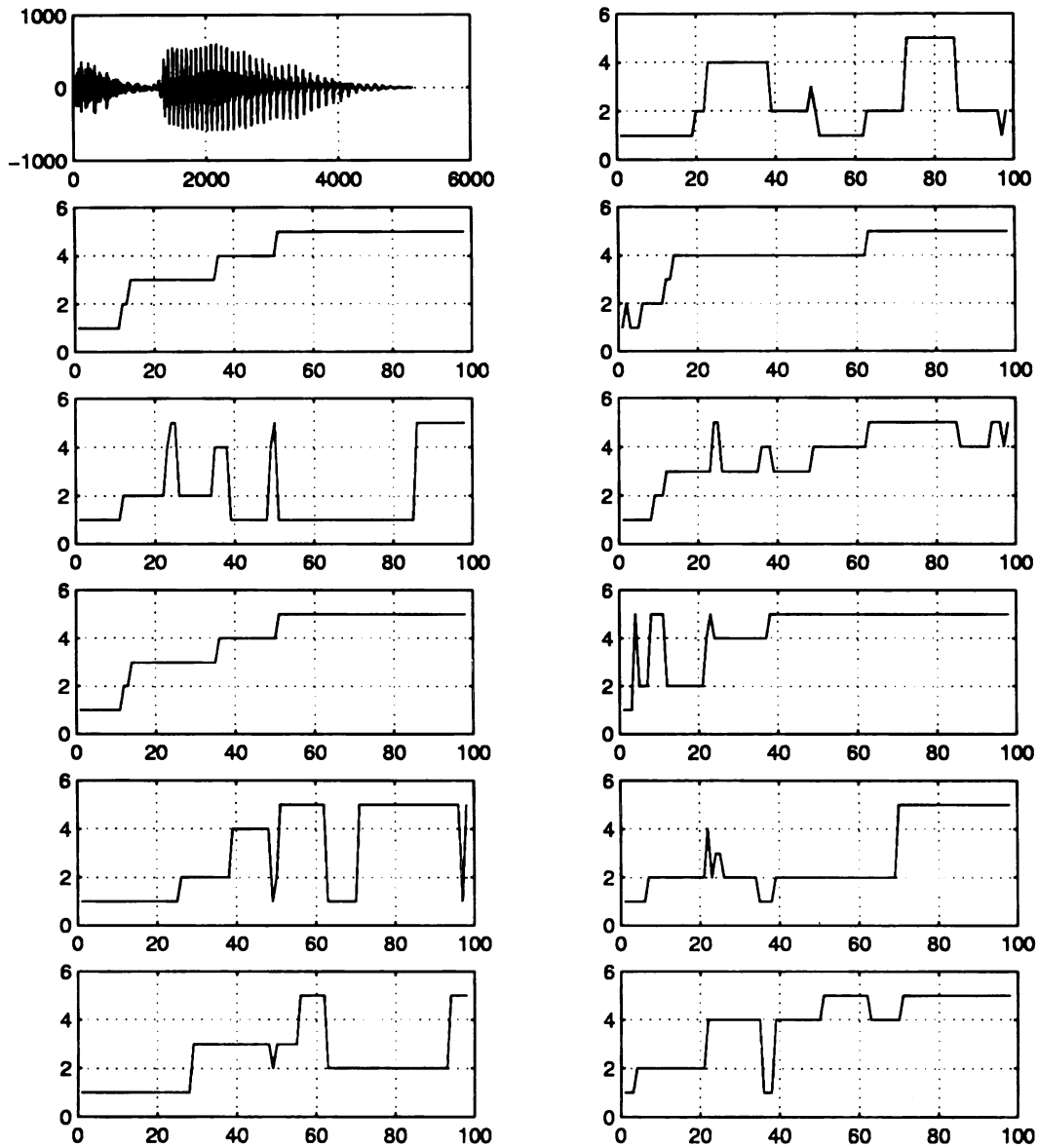

Figure 4.2: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i), \quad i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "two." Note that each graph represents a different "i" except top two figures in the left column. The tests employ resubstitution.

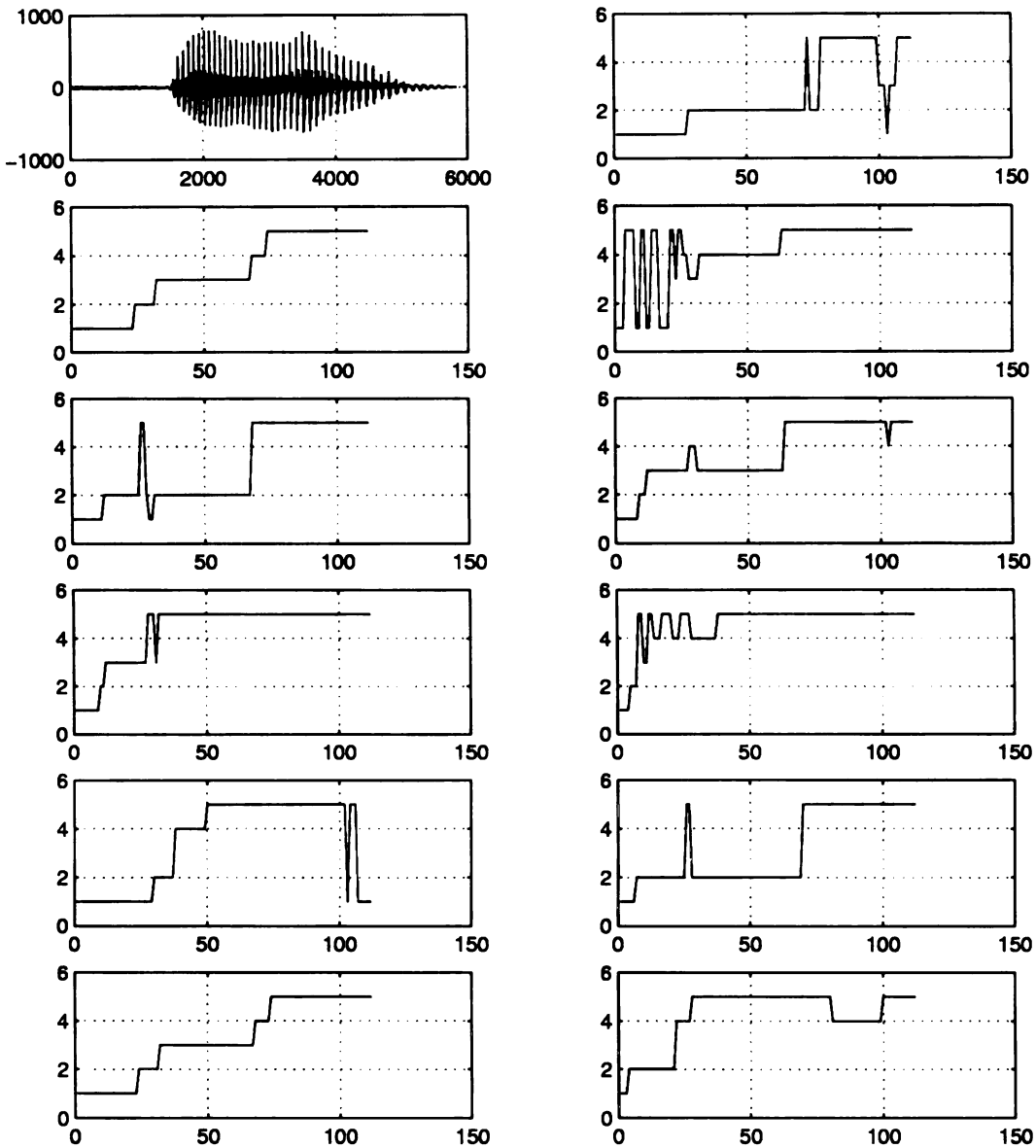

Figure 4.3: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i), \quad i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "four." Note that each graph represents a different "i" except top two figures in the left column. The tests employ resubstitution.

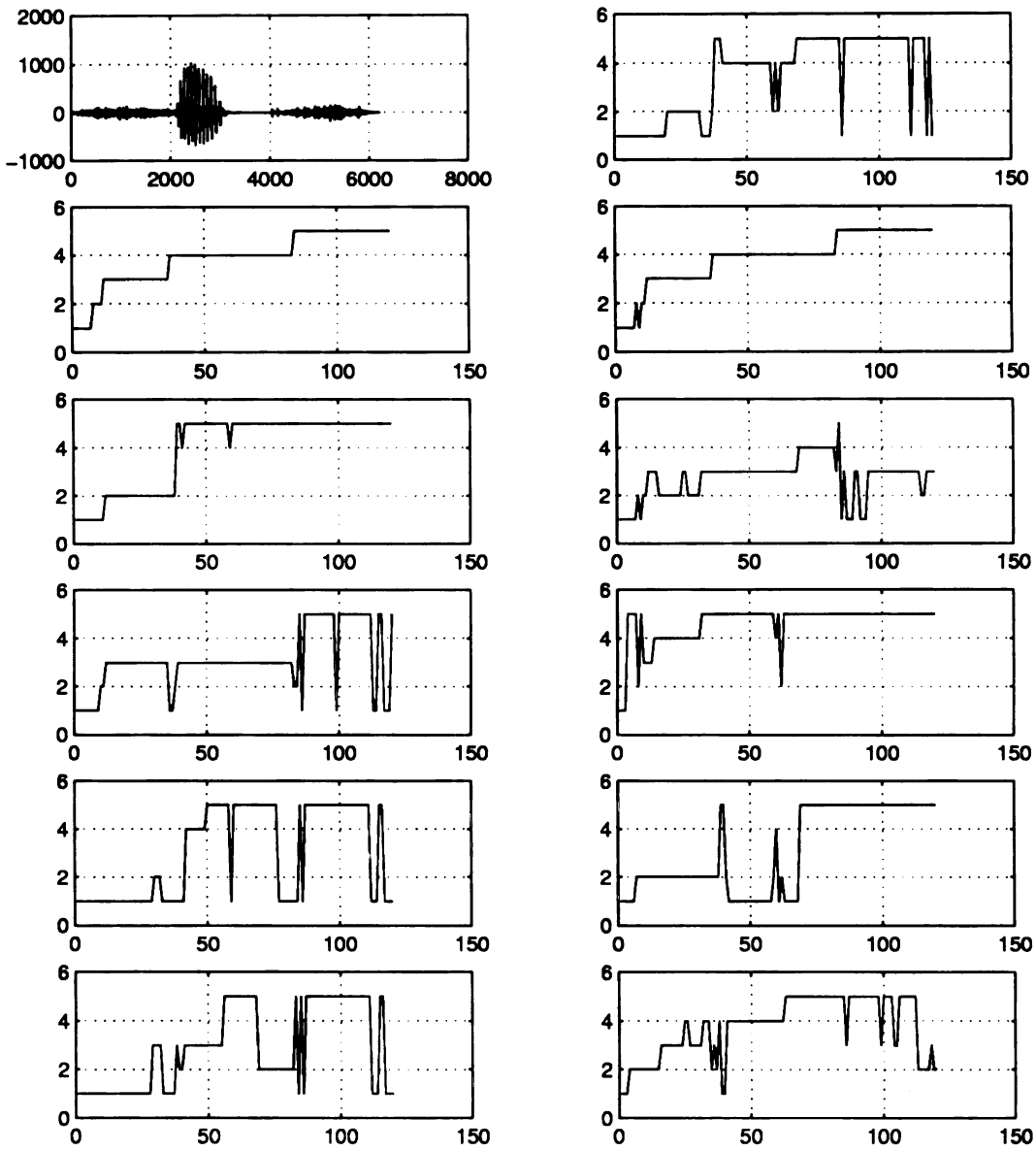

Figure 4.4: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i), \quad i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "six." Note that each graph represents a different "i" except top two figures in the left column. The tests employ resubstitution.

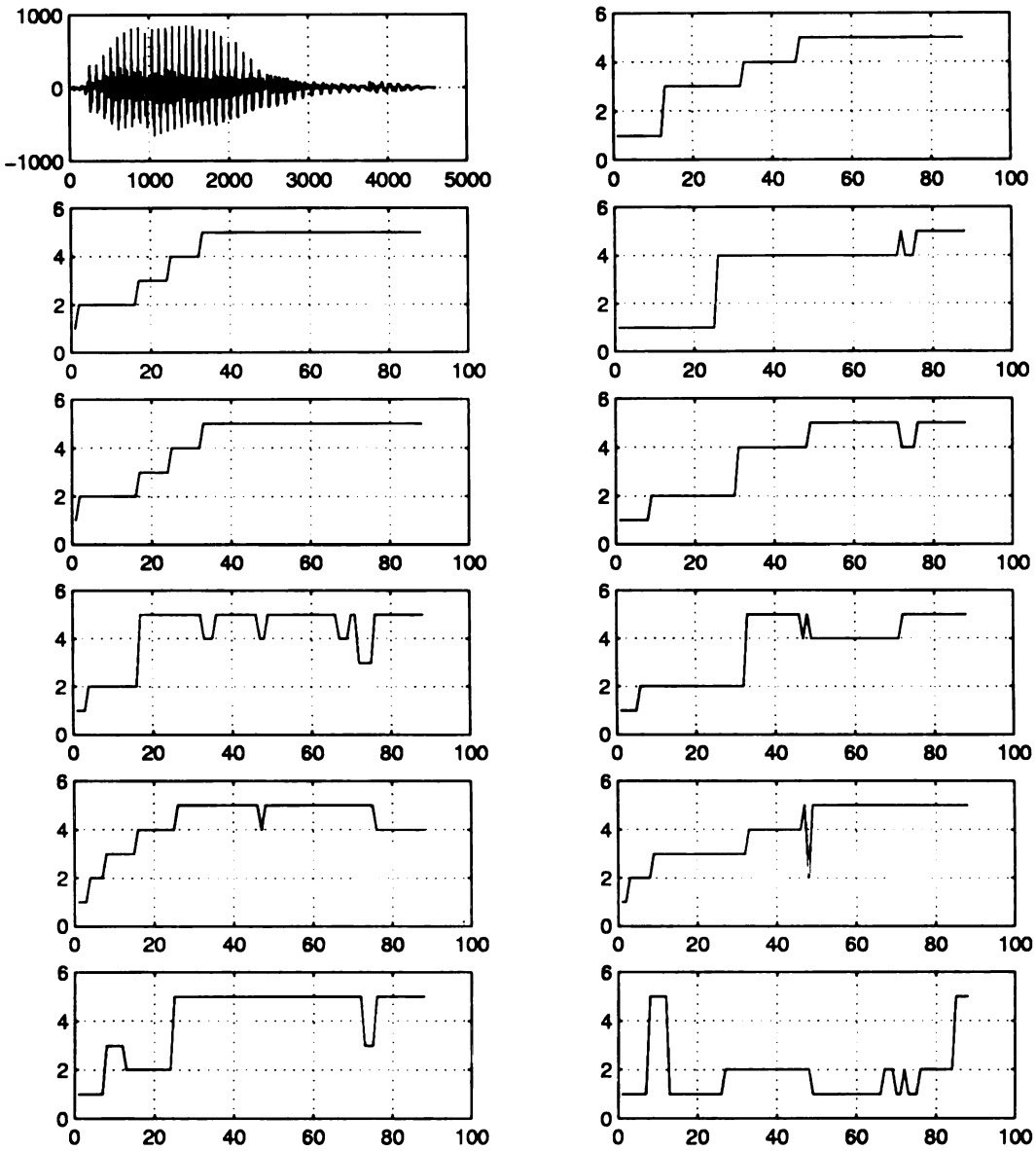

Figure 4.5: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i), \quad i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "one." Note that each graph represents a different "i" except top two figures in the left column. The tests employ leave-one-out.

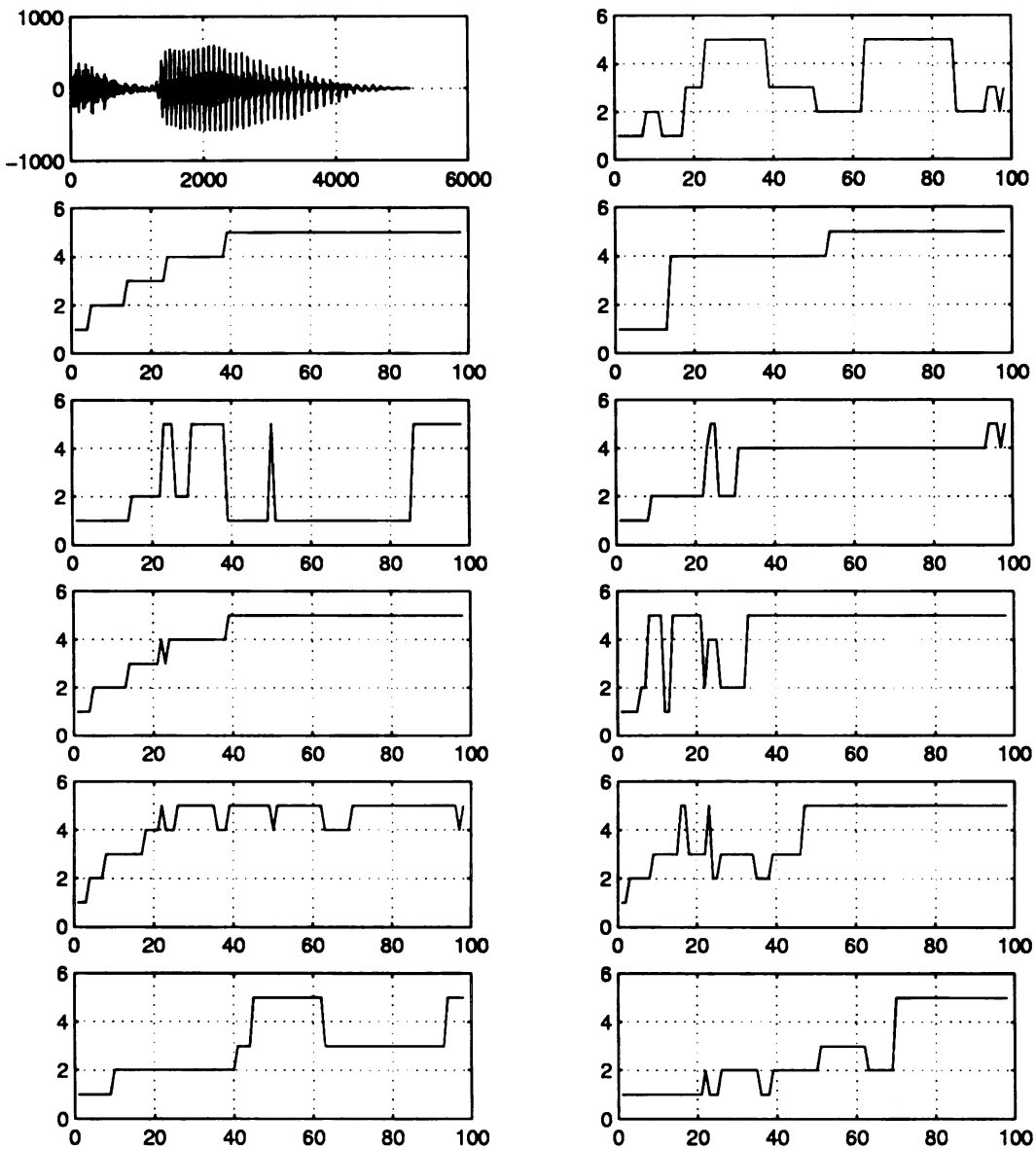

Figure 4.6: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i)$, $i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "two." Note that each graph represents a different "i" except top two figures in the left column. The tests employ leave-one-out.

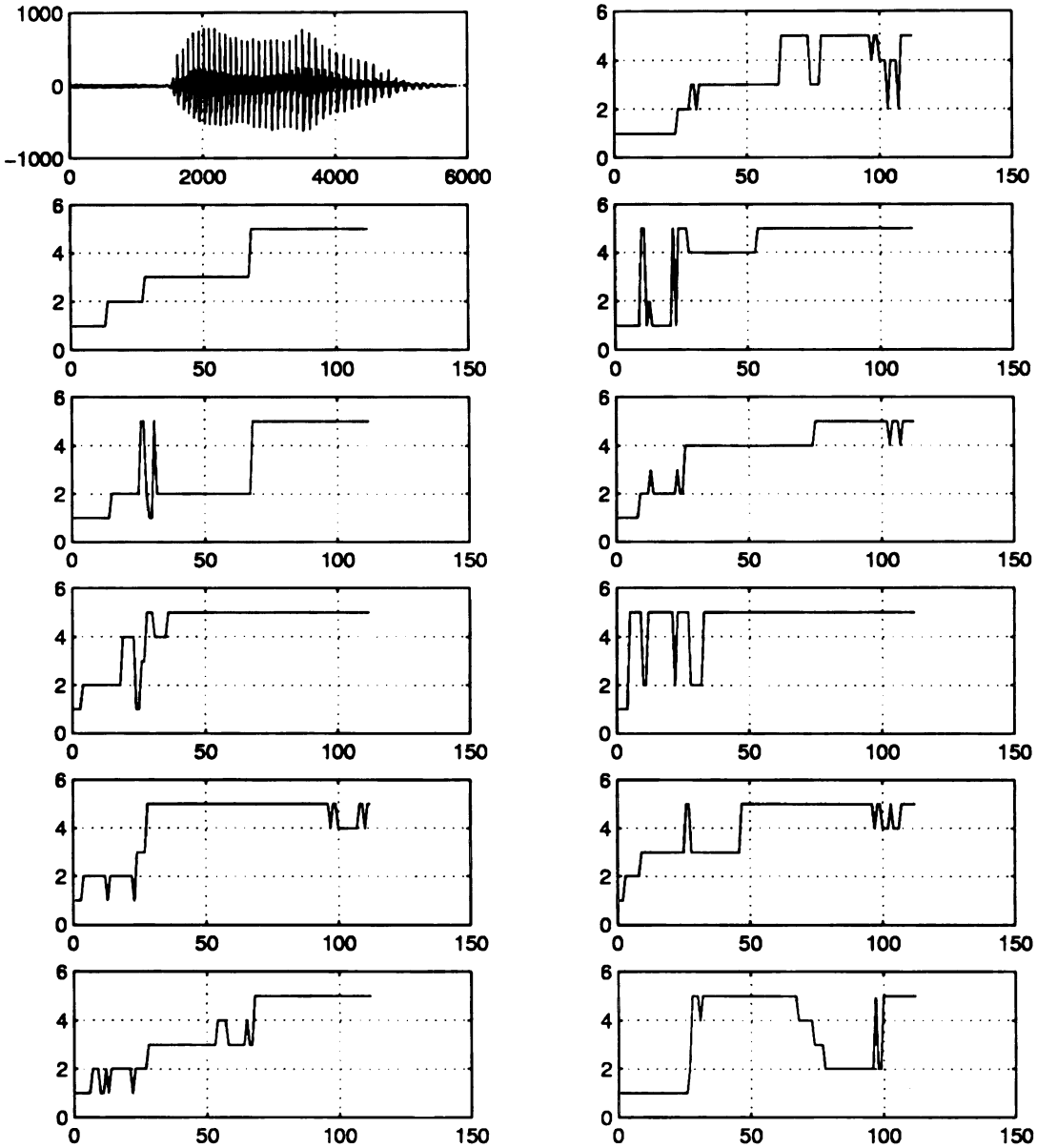

Figure 4.7: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i)$, $i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "four." Note that each graph represents a different "i" except top two figures in the left column. The tests employ leave-one-out.

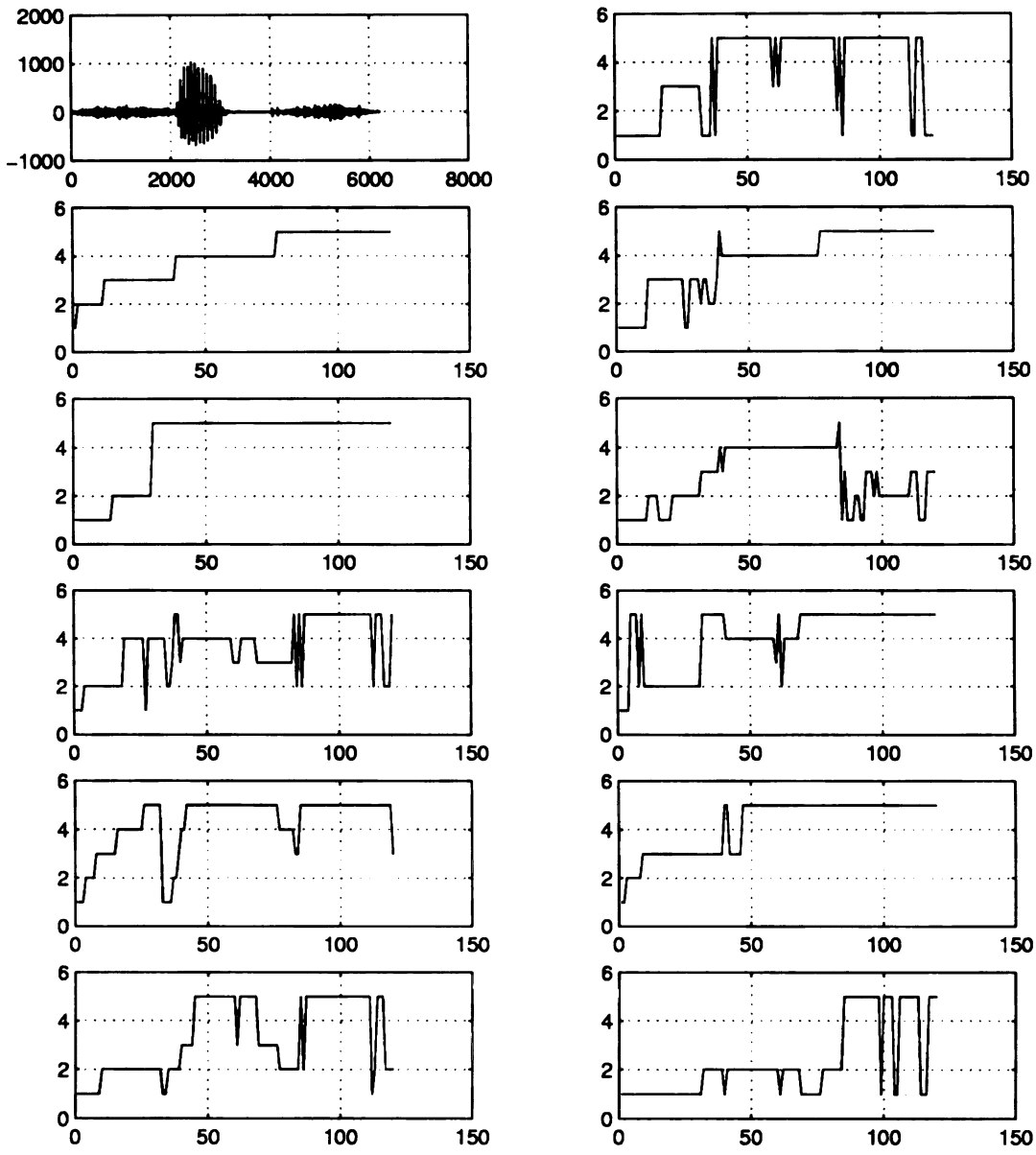

Figure 4.8: State search results from the conventional Viterbi and $Q^* = \prod_t^T q_t^* = \prod_t^T \arg\max_{q_t} P(O_t, q_t \mid \mathcal{M}_i), \quad i = 1, \ldots, 10$ in a five-state Bakis HMM of a spoken word "six." Note that each graph represents a different "i" except top two figures in the left column. The tests employ leave-one-out.

one-out method. In particular, the state search result for the correct word with a resubstitution method adheres more closely to the Bakis constraints than with leave-one-out test.

- Roughly speaking, state sequence results using (4.1) and (4.2) appear to be useful indicators to pre-filter the correct utterances for this limited vocabulary. This implies that, with the likelihood values, but also by examining the sequence, it is possible to sort out the possible "candidate" utterances from the utterance data base. Under a Bakis condition, for example, the state sequence needs to be monotonically non-decreasing. After applying (4.2) and (4.3) to the testing utterances, the correct utterance is one of the utterances in which the computed state sequence is closely consistent with the Bakis topology. Experiments with larger vocabularies are needed to confirm that this is a general phenomenon.

- The global shape of the state sequences found by the conventional Viterbi algorithm and ML criterion using (4.2) and (4.3) for the correct word are similar to one another. Even though there are some short-term peaks which represent jumps to different states and return to the previous states occurring over short time durations, it is not difficult to estimate a possible plausible state sequence in light of Bakis topology by removing such short-term peaks.

- When the correct speech utterance is evaluated, fluctuation in the state sequences are relatively scarce.

At a glance, the proposed criterion does not appear to provide a reasonable criterion for finding the "legitimate" state sequence since the scheme does not explicitly impose constraint on Q^* composed of $q_t^*, \quad t = 1, \ldots, T$ according to a Bakis constraint. However, the state sequence by (4.2) is close to the conventional Viterbi search result without "explicit" constraint of state transition paths.

The implications of those results can be summarized as follows: First, we know that the TIA HMM relies on individual state probabilities at each time to dissuade illegal paths, and it implicitly constitutes available state path information. Thus, the evaluation through the TIA HMM provides a reasonable measure to classify the utterances. Similarly to the evaluation of the HMM by the TIA HMM, criterion (4.2) implicitly has information about a possible state sequence. Like the TIA HMM, criterion (4.2) only indirectly controls the overall state paths.

Next, the results implicitly show the relative significance of $\boldsymbol{A}$ and $\boldsymbol{B}$. In the previous developments, we have assessed the relative significance of $\boldsymbol{A}$ and $\boldsymbol{B}$ in the HMM scoring process. We have dealt with this problem from various points of view in the previous chapters. Added to such attempts, the approach to find an appropriate state sequence can also help to show the relative significance of $\boldsymbol{A}$ and $\boldsymbol{B}$. As a consequence of the various approaches, therefore, it is not difficult to infer that most of the information of the training utterances is concentrated in the elements of $\boldsymbol{B}$, rather than $\boldsymbol{A}$, in the F-B or Viterbi reestimation algorithm.

It is concluded that criterion (4.2) and (4.3) provides a simple and fast way to find a meaningful state sequence of a correct speech utterance approximately without a Viterbi criterion.

4.2 State Sequence Based on "Acoustic Distance"

In this section, a new state search technique using recursive Viterbi search based on an "acoustic" distance is presented. There are several advantages of applying this technique to find an state sequence for a speech utterance. They are discussed here, together with a basic idea and corresponding algorithm. Some results in a practical application will be presented and those results are compared to the results from the conventional Viterbi method.

4.2.1 Introduction

One of the original motivations for using HMMs in speech recognition is the apparent congruity between the speech production process and the mathematical dynamics of the model. Human speech production can be approximately modeled as a process of dynamically positioning the speech system "articulators" into physical "states" which correspond to resulting acoustic outputs. The acoustic manifestations of these states are "observable" to the listener, but the physical states are "hidden." Accordingly, the states of a HMM are often thought of, to a first approximation, as representing distinct acoustical phenomena in the utterance, such as a vowel sound in a word or a transition between phonemes in a word. In fact, the number of states in a model is sometimes chosen to correspond to the expected number of such phenomena. For example, if an HMM is used to model a phoneme (rather than a complete word), then three states might be used – one to capture the transition on either end of the phoneme, and one for the steady-state portion.

However, the HMM organizes itself to maximize an analytic criterion (usually a ML), and not necessarily to correspond to some preconceived acoustic structure. In fact, our work has shown that the conventional ML approaches (Baum-Welch and Viterbi) frequently yield model structures which clearly exhibit little relationship between waveform acoustics and HMM states. See Fig. 4.9 for an illustration. A more "global" view of this phenomenon has led researchers, notably Ostendorf and colleagues [61], to seek "segment based" models of the waveform that are more meaningfully associated with regions of acoustic coherence in the speech.

In this work, we present a simple HMM decoding algorithm which seeks a meaningful state sequence by finding an acoustic similarity among observation symbol so that an acoustic meaningfulness for the state can be achieved. It is a new and simple state searching technique for a given observation sequence using only a distance information within symbols of an utterance so that a state sequence could be consistent

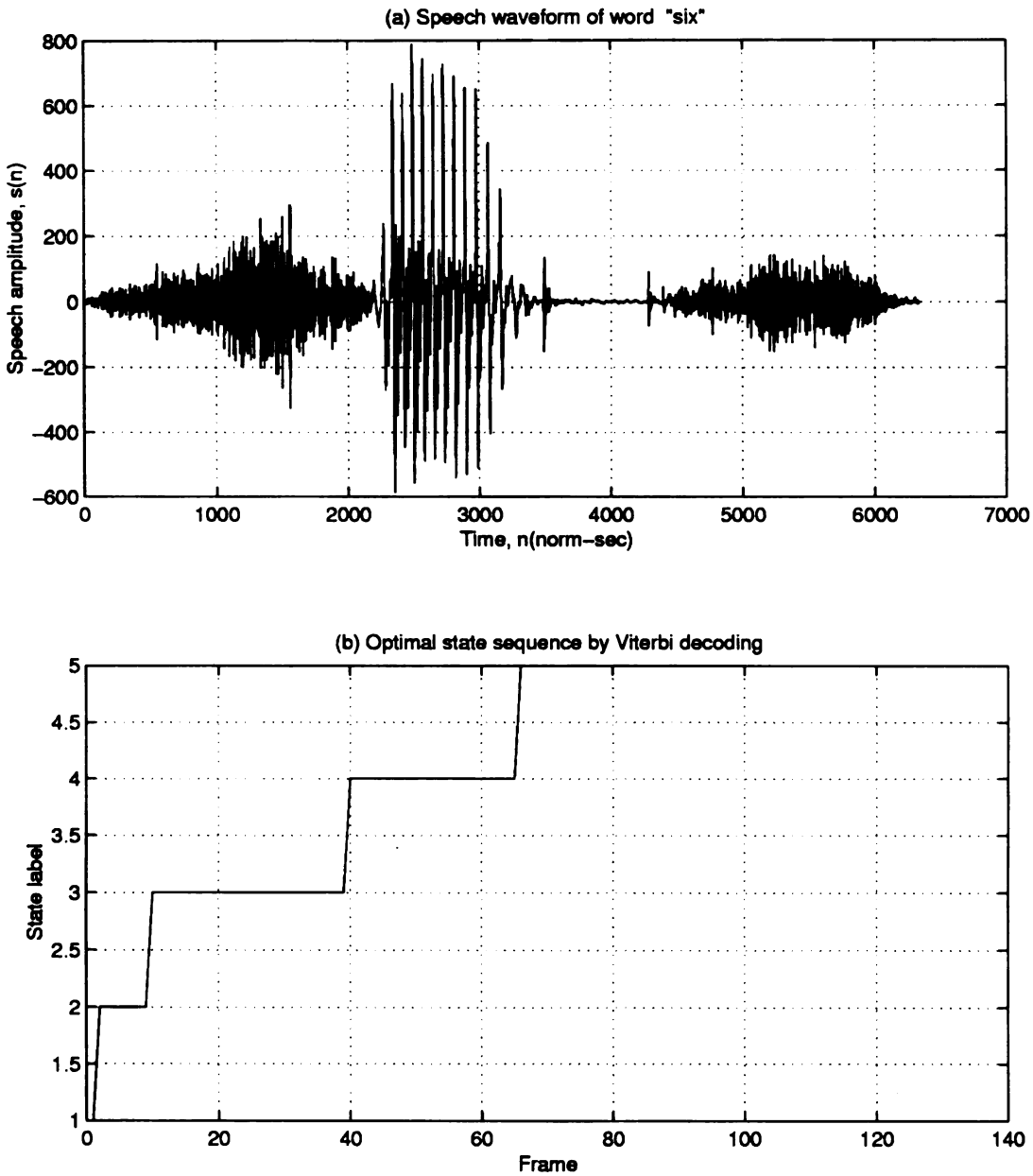

Figure 4.9: State segmentation resulting from conventional ML (Viterbi) training of a five-state Bakis HMM for the utterance "six." The resulting segmentation is not coherent with the physical dynamics of the speech.

with the physical characteristics of a speech waveform.

The work inherently begs the question as to why the conventional HMMs do not exhibit more coherence between the "acoustic states" of the speech, and the analytical states of the model. We provide both qualitative and analytical analyses of this question and also suggest some implications for HMM performance.

4.2.2 The Concept

Suppose that there is a sequence of speech samples along the time axis and the task is to find a state assignment that explains their production in an optimal and meaningful way. From the modeling point of view, the desirable state sequence is such that each state is a set of entities (symbols) that are acoustically similar. This general concept is the basis for clustering algorithms [10]. Using ideas akin to a clustering procedure, we seek an algorithm with which to find an state sequence for a given speech utterance.

Generally, clustering techniques are based on the heuristic argument that samples representing the same cluster should be "close" to one another in the vector space and "far" from vectors representing other clusters. The underlying assumption is that the feature vector representing a sample is appropriate and efficient in capturing similarity among exemplars. The most commonly used clustering strategy is based on the minimum squared-error criterion where squared-error amounts to the distance between a sample and the centroid of a cluster in the vector space. The general objective is to obtain that partition which, for a fixed number of clusters, minimizes the total squared-error. It is known that minimizing squared-error, or within-cluster variation, is equivalent to maximizing the between-cluster variation [10]. In general, a clustering method employs an iterative algorithm to optimize a clustering criterion function. Various criteria have been suggested in the literature, but among these, the family of criterion functions quantifying the average affinity of feature vectors to cluster representatives have proved to be most useful [18].

In a discrete HMM system, let $\mathcal{O} = \{O_1, O_2, \ldots, O_T\}$ be a sequence of a quantized speech symbol string for which we are interested in finding a meaningful state sequence. For each O_t, $1 \leq t \leq T$, there exists a corresponding M-dimensional feature vector $\mathbf{x}_t$ in the vector space and thus we have a set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T\}$ associated with $\mathcal{O}$. In fact, $\mathbf{x}_t$ amounts to a vector representing a centroid of a certain cluster in the codebook since index O_t is the result of quantization of a signal. Frequently *mel*-cepstral coefficients are used for elements of $\mathbf{x}_t$ of when processing a speech signal. Also, let $\mathcal{S} = \{1, 2, \ldots, N\}$ be a finite set of sequential natural numbers, each representing a state. The task is to associate $\mathcal{X}$ with a sequence $Q = (q_1, q_2, \ldots, q_T), \quad q_t \in \mathcal{S}$, in a meaningful way based on a given optimization criterion.

For this task, suppose that we initially have N partitions for $\mathcal{X}$ and let us denote them as $\mathcal{C} = \{c^{(1)}, c^{(2)}, \ldots, c^{(k)}, \ldots, c^{(N)}\}$. Therefore, $c^{(k)}$ is also a set and it has $n^{(k)}$ entries as

$$c^{(k)} = \{x_{\sum_{j=1}^{k-1} n^{(j)}+1}, \ldots, x_{\sum_{j=1}^{k} n^{(j)}}\} \tag{4.7}$$

so that

$$\sum_{i=1}^{N} n^{(i)} = T. \tag{4.8}$$

Simply, let us have an initial state segmentation such that the entire symbol string is divided into N segments of approximately equal lengths. How the initial segmentation effects the performance will be briefly discussed later. Also, let $\mathcal{G} = \{m^{(1)}, m^{(2)}, \ldots, m^{(N)}\}$ be a set representing the centroids of clusters $c^{(k)}$, $k = 1, \ldots, N$. Let the distance between x_t and $m^{(k)}$ of cluster k be denoted by $d(x_t, m^{(k)})$. Here $k \in \mathcal{S}$ so that eventually a sequence of states is denoted in terms of sequence of clusters.

Then, we have three sets as $\mathcal{G}$, $\mathcal{X}$, and $\mathcal{D}$. Where $\mathcal{D}$ is defined as

$$\mathcal{D} = \{d(x_t, m^{(k)})\}_{1 \le t \le T,\ 1 \le k \le N}. \tag{4.9}$$

Note that since the initial number of clusters may not be the number of clusters, the condition of merging of clusters may be imposed in the clustering algorithm. In that case, the number of clusters can be less than N.

Now the task is to seek a meaningful state sequence $Q^* = \{q_1, q_2, \ldots, q_T\}$, $q \in \mathcal{S}$ under an optimality criterion.

Using the clustering algorithm, we seek a sequence Q^* based on the criterion

$$Q^* = \arg\min_{Q} \sum_{t=1}^{T} d(x_t, m^{(q_t)}) \tag{4.10}$$

$$= \arg\min_{Q} \boldsymbol{d}_{[1,T]}, \tag{4.11}$$

where $Q = \{q_1, q_2, \ldots, q_T\}$, $\boldsymbol{d}_{[1,T]} = \sum_{t=1}^{T} d(x_t, m^{(q_t)})$. It is known that the Euclidean distance between two cepstral vectors representing features is a reasonable measure of spectral similarity in the models [4]. Hence, let $d(x_t, m^{(q_t)}) = \|x_t - m^{(q_t)}\|_2$. Additionally, the constraint

$$q_1 \le q_2 \le \ldots \le q_{T-1} \le q_T \tag{4.12}$$

is required for the Bakis model. This algorithm iterates until Q^* converges to a stable state sequence. The algorithm is based on the fact that there is high metric similarities between the components within the same cluster and high metric dissimilarities between different clusters.

In developing this technique, we assume a discrete HMM and thus a speech signal is assumed to be quantized. However, in fact, $\mathbf{x}_t$ can be replaced with the unquantized cepstral feature vector in this development since the algorithm is based on distance

information between feature vectors. Usually the speech frame vectors are quantized by the *LGB* algorithm [4] which is based on the k-means algorithm. Basically, k-means algorithm is one of the ways of *hierarchical clustering* [4]. In a hierarchical clustering procedure like k-means, theoretically there does not exist ordered relations between objects that are located at different places in the same layer in the hierarchical tree. However, in our development, we use a *partitional clustering* method which is used frequently in engineering and science for problems in which single partitions are important. Therefore, it is significant to check the proximities among symbols in the codebook which are encoded by hierarchical clustering.

Figure 4.10, for example, shows the Euclidean distances between the features representing for different symbols such as "zero," "32," "64," and "96" and the rest of the features representing the symbols derived from spoken digits with cepstral features quantized to 7 bits (128 levels) by LGB. For the other symbols other than the above four symbols, we see similar pictures to Fig. 4.10.

The figure shows that although the distances are not completely ordered as the symbols assigned to the feature are ordered, the Euclidean distance is another meaningful indicator to show the proximities among symbols. Here symbols are encoded by hierarchical clustering. Therefore, the quantized symbols may be classified according to the partitional clustering method using the Euclidean distance between features.

The algorithm for finding a meaningful state sequence based on (4.10) through (4.12) is as follows:

1. Initialization

   One of N states is initially assigned to each feature in $\boldsymbol{\mathcal{X}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$. $\boldsymbol{\mathcal{X}}$ can be divided into N approximately equal segments. Each components of a given segment is assigned the integer index of its associated segment, say,

$$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_1}\} \mapsto 1$$

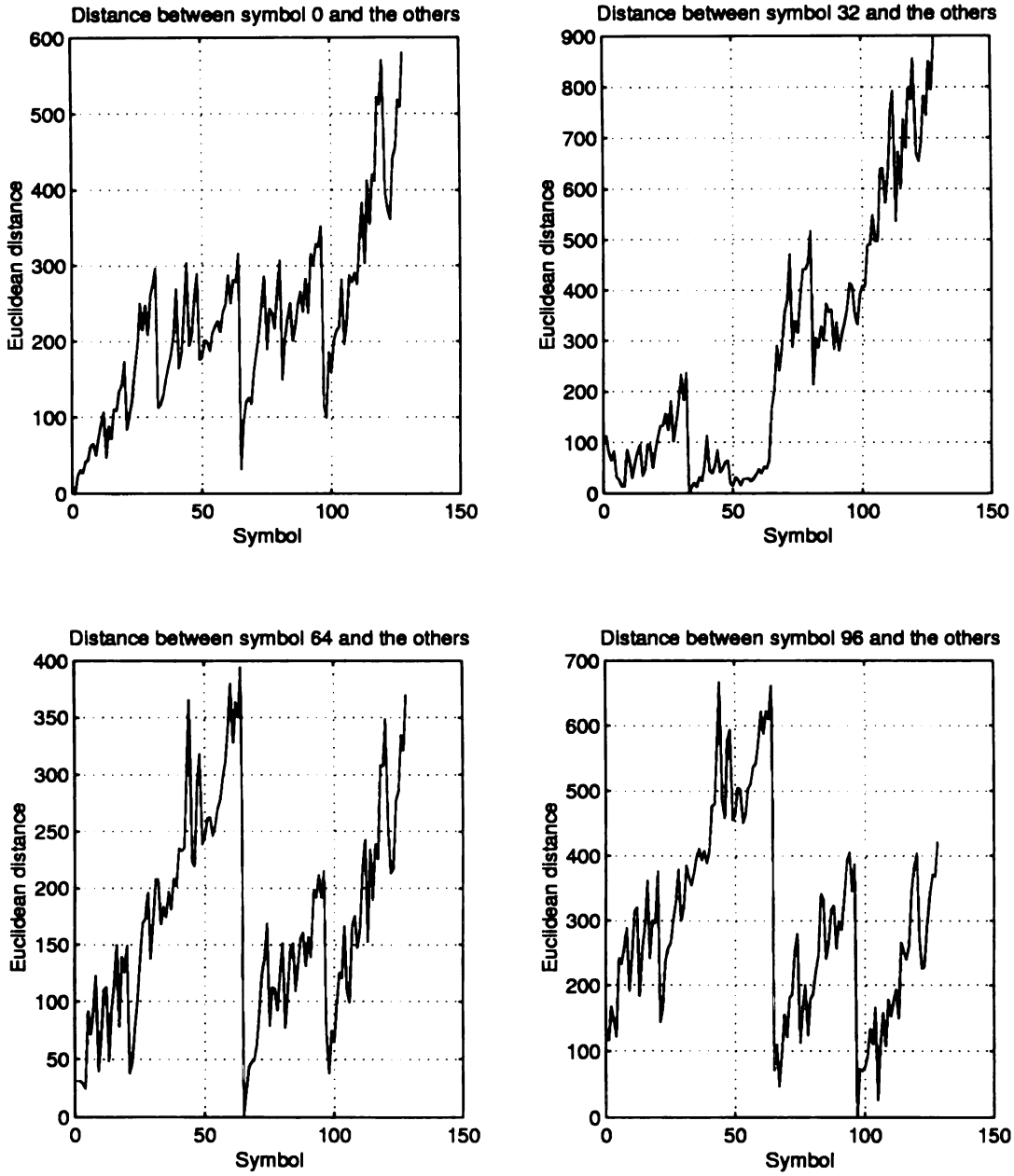

Figure 4.10: Euclidean distances between four different symbols (symbol "zero," "32," "64," and "96") and the rest of symbols indexed along the abscissa in the codebook. Each symbol represents a centroid of the cluster in the feature vector space.

$$\{\mathbf{x}_{n_1+1}, \mathbf{x}_{n_1+2}, \ldots, \mathbf{x}_{n_2}\} \mapsto 2 \tag{4.13}$$

$$\vdots$$

$$\{\mathbf{x}_{n_{N-1}+1}, \mathbf{x}_{n_{N-1}+2}, \ldots, \mathbf{x}_T\} \mapsto N.$$

Also, let

$$\mathcal{G}_1 = \{m^{(1)}, m^{(2)}, \ldots, m^{(N)}\} \tag{4.14}$$

be the set of centroids of the initial segments. Therefore, the elements of $\mathcal{G}_1$ are given by

$$m^{(j)} = \frac{\sum_{i=n_{j-1}+1}^{n_j} \mathbf{x}_i}{n_j - n_{j-1}} \tag{4.15}$$

for $j \in [1, N]$ with $n_0 = 0$ and $n_N = T$. Also, if Bakis topology is concerned, which is often employed in speech recognition, let

$$q_1^* = 1, \quad q_T^* = N. \tag{4.16}$$

2. Recursion

   For $l = 2, 3, \ldots,$

   For $t = 2, 3, \ldots, T-1,$

$$x_t \mapsto q_t^* \tag{4.17}$$

where

$$q_t^* = \arg\min_{q_t} d(x_t, m^{(q_t)}) \quad \text{and} \quad q_{t-1}^* \leq q_t^*. \tag{4.18}$$

Next t

Recompute $\mathcal{C}, \mathcal{G}, \mathcal{D}$ with reassigned elements. If $\mathcal{G}_l = \mathcal{G}_{l-1}$, go to step 3. Otherwise, Next l.

3. Termination

$$Q^* = \{q_1^*, q_2^*, \ldots, q_T^*\} \tag{4.19}$$

Note that except for the initial cluster, the only required data for the recursive Viterbi search based on k-means clustering comprise the set of distances between centroids and a features.

To give the algorithm a probabilistic flavor, consider a mapping which transforms distance measure to a probabilistic measure by using the mapping

$$P = f(d) = e^{-d} \tag{4.20}$$

so that

$$P(q_t = i) = e^{-d(x_t, m^{(i)})}. \tag{4.21}$$

This transformation makes the above algorithm more like a ML formulation of state search algorithm similar to the conventional Viterbi search which is based on the ML criterion. In (4.20), it is straightforward to see that as $d(x_t, m_i) \to \infty$, the probability that q_t is i approaches zero, while conversely, as $d(x_t, m_i) \to 0$, the probability that q_t is i approaches unity. This is acoustically reasonable because in the high dimensional vector space, a small distance between two cepstral features vectors implies that the corresponding acoustic frames are very similar acoustically and, thus, should be

assigned to the same state. Therefore, (4.10) with constraint (4.12) can be written as

$$\begin{aligned} Q^* &= \arg\max_{Q} \prod_{t=1}^{T} P(q_t = i_t) \\ &= \arg\max_{Q} \prod_{t=1}^{T} e^{-d(x_t, m^{(q_t)})} \\ &= \arg\max_{Q} e^{-\sum_{t=1}^{T} d(x_t, m^{(q_t)})} \\ &= \arg\max_{Q} e^{-\boldsymbol{d}_{[1,T]}} \end{aligned} \tag{4.22}$$

where $i_t \in \{1, 2, \ldots, N\}$ for all t. This expression could be useful to assess the relationship between the conventional Viterbi search technique and the technique above. We will have more to say later about the relation between the conventional Viterbi technique and the recursive Viterbi search technique suggested here.

4.2.3 Recursive Viterbi Search Based on k-Means

The sequence Q^* can be computed by a conventional k-means algorithm [9]. However, in the case of speech signals where time ordering of speech samples is important associated with a state sequence constrained by Bakis topology for example, the conventional Viterbi search technique is an appropriate way for finding a meaningful state sequence. Therefore, let us apply the conventional Viterbi search technique in finding Q^* in our problem. The Viterbi searching is engineered to find an path which satisfies (4.10) and (4.12) under the Bakis topology. Here, note that in contrast to an ordinary Viterbi decoding technique, the proposed approach requires Viterbi technique to be applied iteratively until Q^* converges to a stable state sequence. This is because the algorithm requires arbitrary initial clusters and the algorithm is based on the squared-error between a feature and a centroid. Therefore, the resulting state sequence may be different at each iteration depending on the members of clusters.

This recursive Viterbi search employs k-means to find a meaningful state sequence

using the distance information among symbols. The algorithm is described in the following way. Here, we need two more variables $d_t(j), \psi_t(j)$ for this development. Here $d_t(j)$ stands for a sum of distances between the feature and a centroid of cluster from $t = 1, \ldots, t$ over the bestpath until it reaches state j at time t. $\psi_t(j)$ represents a state at time $t-1$ which corresponds to $d_t(j)$.

1. Initialization

   Create initial clusters as in (4.13), and let

$$\boldsymbol{C}_1 = \{c_1^{(1)}, c_1^{(2)}, \ldots, c_1^{(N)}\}. \tag{4.23}$$

   The centroids of N-clusters comprising the set $\boldsymbol{\mathcal{G}}_1 = \{m_1^{(1)}, m_1^{(2)}, \ldots, m_1^{(N)}\}$ are given by (4.15). Furthermore, let $q_1^* = 1, \quad q_T^* = N$ as before. Additionally, the initial values of the variables $d_t(i)$ and $\psi_t(j)$ are given by

$$d_1(1) = 0, \quad d_1(i) = \infty \text{ for } 2 \leq i \leq N \tag{4.24}$$

$$\psi_1(j) = 0, \quad \forall j = 1, \ldots, N. \tag{4.25}$$

2. Recursion

   For $l = 1, 2, \ldots,$

   For $t = 2, 3, \ldots T$, and for $j = 1, 2, \ldots N$,

$$d_t(j) = \min_{1 \leq i \leq N} \{d_{t-1}(i) + d(x_t, m_{l-1}^{(j)})\} \tag{4.26}$$

$$\begin{aligned} \psi_t(j) &= \arg \min_{1 \leq i \leq N} \{d_{t-1}(i) + d(x_t, m_{l-1}^{(j)})\} \\ &= \arg \min_{1 \leq i \leq N} \{d_{t-1}(i)\} \end{aligned} \tag{4.27}$$

Next t

$$d_{[1,T]}^{min} = \min_{1 \le j \le N} d_T(j) = d_T(N) \tag{4.28}$$

$$q_T^* = N. \tag{4.29}$$

Backtracking:

$$q_t^* = \psi_t(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1. \tag{4.30}$$

$$Q_l^* = \{q_1^*, q_2^*, \ldots, q_T^*\} \tag{4.31}$$

Recompute $\boldsymbol{C}_l = \{c_l^{(1)}, c_l^{(2)}, \ldots, c_l^{(N)}\}, \boldsymbol{G}_l = \{m_l^{(1)}, m_l^{(2)}, \ldots, m_l^{(N)}\}$. If $\boldsymbol{G}_l = \boldsymbol{G}_{l-1}$, go to step 3. Otherwise, Next l.

3. Termination:

$$Q^* = Q_l^* = \{q_1^*, q_2^*, \ldots, q_T^*\} \tag{4.32}$$

The probabilistic version of this algorithm would use (4.20) in place of the direct use of the metric.

4.2.4 Experimental Results

In contrast to the conventional Viterbi technique or the state search with a criterion (4.2), the proposed technique does not require any specific *a priori* knowledge like $\{\boldsymbol{A}, \boldsymbol{B}\}$ from the training utterance set. Additionally, this technique does not have mismatch problem between the training and testing data. Only an initial segmentation information is required. Therefore, this technique can be applied universally to find a state sequence for a speech utterance.

The Relationship between Likelihoods and the State Path in the conventional Viterbi Technique

First, let us consider the effects of an initial values of the two significant matrices in the F-B HMM. The state segmentation results using the resubstitution method for the word "six" is shown Fig. 4.11. Each graph displays a state sequence and the corresponding likelihood for a different set of initial values for $\boldsymbol{A}$ and $\boldsymbol{B}$ by the conventional Viterbi search technique. For the numerical purpose, $-\log_{10} P(\boldsymbol{\mathcal{O}} \mid \boldsymbol{\mathcal{M}})$ along with a optimal path is computed.

Figure 4.11 shows the impact of varying the initial values of $\boldsymbol{A}$ and $\boldsymbol{B}$. In addition, it shows the variable relationship between a state sequence and its corresponding likelihood quantity. For example, although the likelihoods of two results are close to each other, the state sequences could be quite different. Also, the state sequence which has largest likelihood (the second figure) may be not coherent with the acoustics of the utterance.

In an extreme case, the state sequence can be poorly matched to the acoustical pattern of the corresponding speech utterance. This is because in the F-B HMM, for example, the likelihood is composed of T-multiplications of pairs of a_{ji} and $b_j(k)$. Under a Bakis topology with N states, $a_{N,N}$ is always one and it is highly probable that $a_{1,1}$ is close to one and it is greater than $a_{i,i}$, $\quad i \in [2, N-1]$. To have large likelihood under T-multiplications of pairs of probabilistic terms lying between zero and one as (2.13), each a_{ji} and $b_j(k)$ needs be as close as to one as much as possible. Therefore, when the HMM is reestimated from the lengthy training utterances, the HMM is, if possible, configured to have many initial (1) state and last (N) state when structured according to the ML criterion in certain case. Accordingly, $\boldsymbol{A}$ and $\boldsymbol{B}$ could be trained to assign as many initial and final states as possible without considering a congruity of the state path with the acoustic "states" of the utterance.

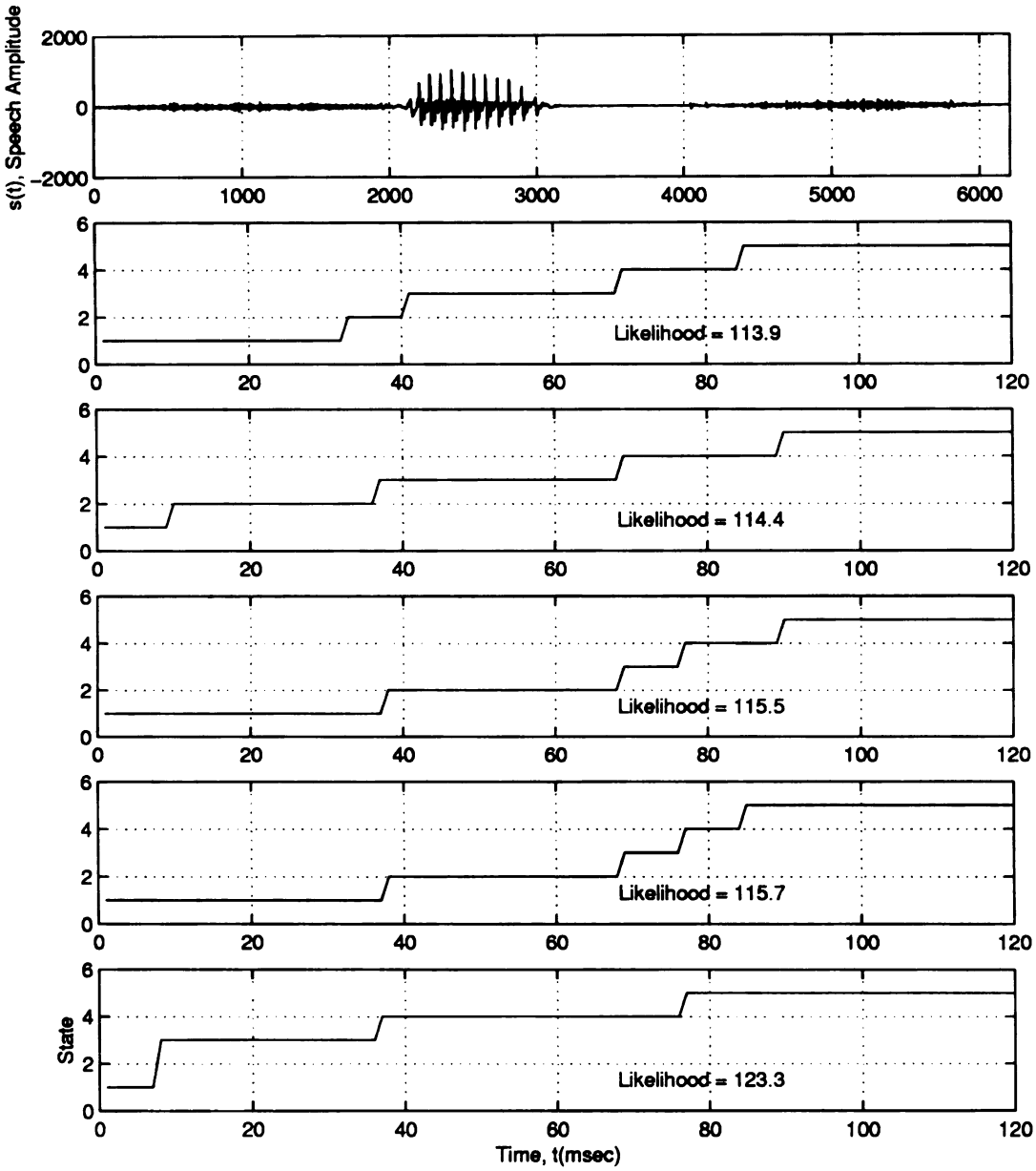

Figure 4.11: State sequence for the spoken word "six" by the Viterbi search technique of a five-state Bakis HMM. Five different sets of initial values have been assigned to $\boldsymbol{A}$ and $\boldsymbol{B}$.

State Search by Recursive Viterbi Based on k-Means

Let us apply the recursive Viterbi search based on k-means clustering to find a meaningful state sequence corresponding to a speech utterance. In particular, the spoken utterances "four" and "six" are considered as examples. Figures 4.12 and 4.13 are the resulting state sequences with the conventional Viterbi technique and recursive Viterbi search based on k-means clustering. The first graph in Fig. 4.12 is the speech waveform for the word "four." The second figure is the result of the conventional Viterbi search technique. The third figure is result of the recursive Viterbi search based on k-means clustering. In addition, we need to compare the likelihoods of the conventional Viterbi and the recursive Viterbi search based on k-means clustering. Since probabilistic likelihood is not a formal measure to the proposed recursive Viterbi search technique, it is necessary to have state sequence information as well as likelihood of the recursive Viterbi search based on k-means clustering in terms of the conventional Viterbi search technique. For this purpose, initially, the recursive Viterbi search based on k-means clustering is applied to all fifteen training utterances to supply a state information. Then, by counting the symbols in each state, two matrices $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ of the HMM are constructed. From these matrices, we can apply the conventional Viterbi technique to find a meaningful state sequence and corresponding likelihood. These are shown in the fourth graph of Fig. 4.12. The shapes of the third and last figures are frequently identical.

For utterance "four," the resulting state sequence by the conventional Viterbi technique and the recursive Viterbi search based on k-means clustering are not very different. This is because there are too many states in the model. However, for word "six," it is apparent that the resulting state sequence by the recursive Viterbi search based on k-means clustering is more consistent with the acoustic properties the utterance. This is because the Viterbi algorithm has been focused on the finding a state sequence which produces a ML criterion rather than considering the acoustic

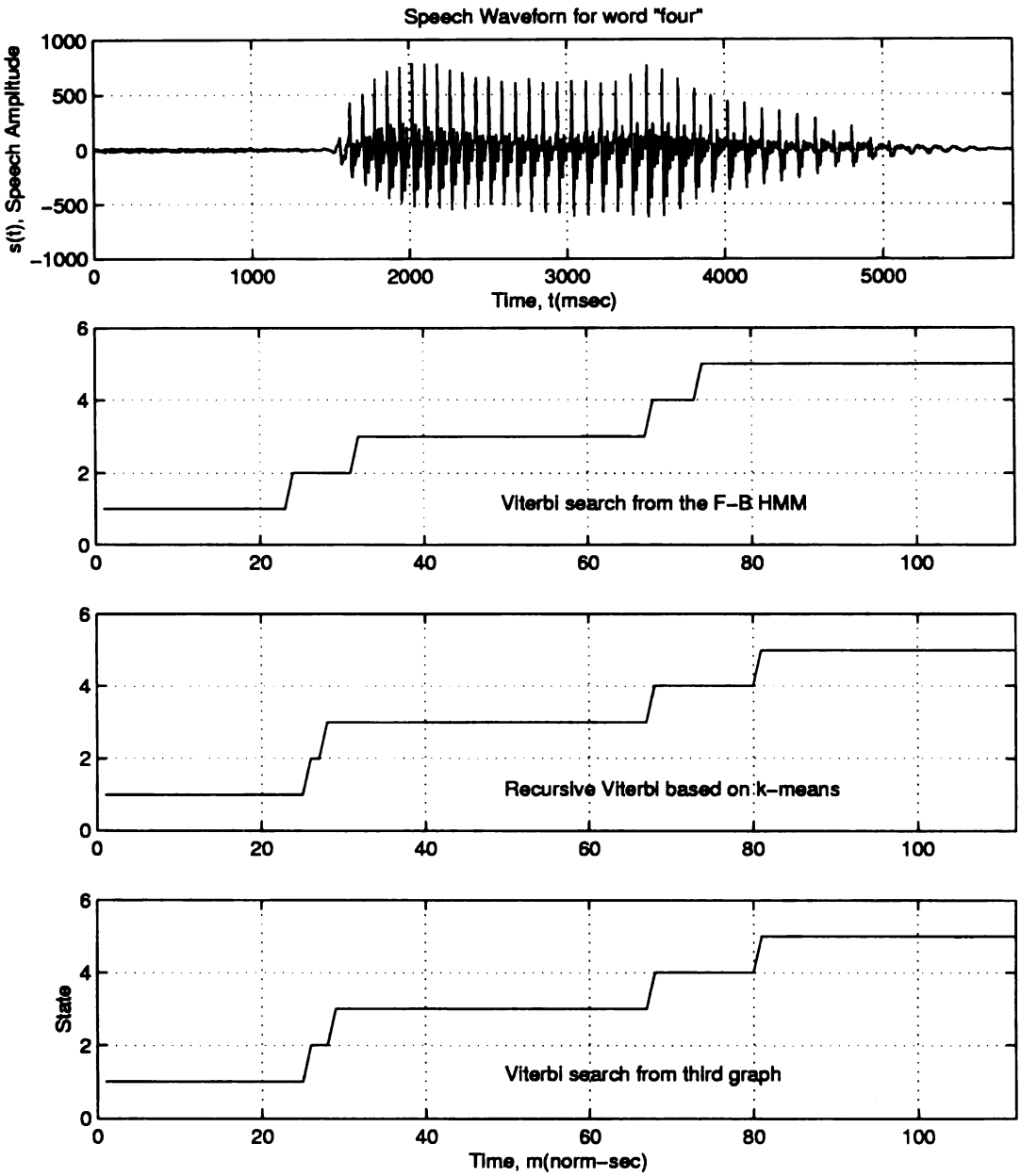

Figure 4.12: State sequences for the spoken word "four" in a five-state Bakis HMM. The second figure is the consequence of the conventional Viterbi search. The third figure is the result of the recursive Viterbi search based on k-means clustering. The 4^{th} graph is the conventional Viterbi search result based on the third graph.

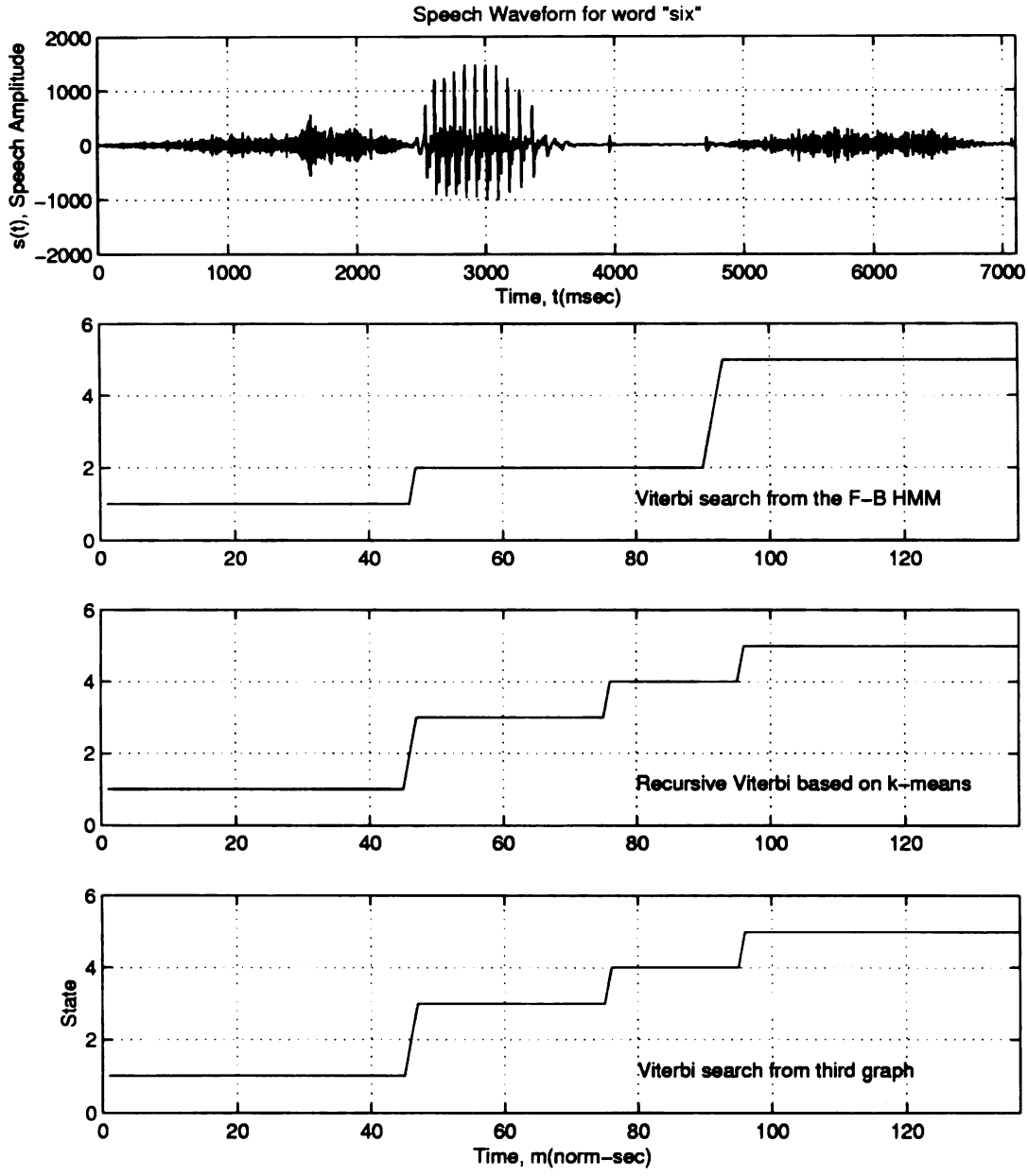

Figure 4.13: State sequences for the spoken word "six" in a five-state Bakis HMM. The second figure is the consequence of the conventional Viterbi search. The third figure is the result of the recursive Viterbi search based on k-means clustering. The 4^{th} graph is the conventional Viterbi search result based on the third graph.

structure of the speech signal. On the other hand, the recursive Viterbi search based on k-means clustering is mainly focused on acoustic features of speech signals.

The likelihood quantities for all fifteen training utterances of "four" and "six" resulting from the conventional Viterbi technique and the recursive Viterbi based on k-means clustering are shown in Table 4.1. Table 4.1 shows that frequently we can

utterance	*"four"*		*"six"*	
	Viterbi	k-means	Viterbi	k-means
1st	70.7	69.5	134.4	131.7
2nd	80.6	74.9	173.4	164.5
3rd	71.2	68.9	150.2	143.4
4th	70.1	65.8	124.9	126.3
5th	55.2	51.7	127.8	122.4
6th	65.9	63.7	120.8	119.0
7th	134.3	106.2	145.3	145.0
8th	79.6	76.1	128.4	129.2
9th	78.6	73.0	138.7	134.0
10th	75.6	74.1	136.4	134.1
11th	84.8	84.4	145.8	138.7
12th	82.7	80.0	118.1	116.5
13th	85.9	83.0	141.6	138.3
14th	71.8	74.3	106.5	103.3
15th	65.2	67.6	135.9	139.5

Table 4.1: Likelihoods of the conventional Viterbi search and the recursive Viterbi search based on k-means clustering.

find a state sequence which not only is acoustically better consistent with a speech utterance, but also has more likelihood than the conventional Viterbi search.

Discussion

We have focused on an HMM training algorithm which seeks to optimize acoustic meaningfulness of the HMM in the sense of minimizing a Euclidean distance of acoustic similarity among observations assigned to given states. This work differs from that of Ostendorf *et al.* [61], not only in its focus on a more "localized" view of

the waveform (frame processing with the explicit goal of training an HMM), but more importantly in the explicit attempt to optimize acoustic state-wise similarity rather than optimally segment the waveform using conventional ML. Similarly to the conventional methods, the new algorithm can also be employed in recognition strategies which assign scores based on acoustic match. The new method inherently provides appropriate dynamic time warping of training and test strings, and is readily modified to optimize over multiple training sequences.

It can also be used to objectively and dynamically guide the selection of the number of model states, the need for state merger, and to assess certain changes in topology on-line. Most importantly, the new method is provably convergent to a local minimum of acoustic mismatch, and regularly provides HMMs with meaningful relationships between states and the acoustic content of the speech that the HMM represents. Also this proposed work is different from the segmental method of states by k-means segmentation [11, 33] since the proposed technique is based on the distance metric between centroids and a sequence of symbols of speech. However, in [11, 33], the segmental method is based on the F-B HMM which has parameter matrices $\{\boldsymbol{A}, \boldsymbol{B}\}$.

Clustering, an unsupervised learning technique, has been widely applied to problems in pattern recognition and classification [9, 10]. In speech signal technology, this technique has been particularly applied in generating a codebook for speech coding, classifying a speaker, and acoustic modeling [4, 11, 12, 13, 14, 15, 33].

First, in contrast to the Viterbi algorithm which requires *a priori* knowledge $\boldsymbol{\mathcal{M}}$, the recursive Viterbi search based on k-means clustering does not require any information. Knowing a state sequence without *a priori* knowledge has a few useful implications. The state information for an utterance can help to exploit and adopt various linear time-invariant system techniques to the speech processing technology by obviating the modeling of the time-varying dynamics.

Next, although the quantized symbols have been used in the development of the algorithm, such quantization is not a requirement of the method. The cepstral feature vector is directly involved in the computations. Therefore, the quantization distortion can be avoided.

Finally, the relation between the conventional Viterbi and the recursive Viterbi search based on k-means clustering may be interesting ultimately although it is not clear from the present study. The hard part of this analysis is to uncover how the distance information between centroids and feature vectors is dynamically distributed in the $\mathcal{M} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. In the F-B HMM training, the dynamics or characteristics of an utterance are transmitted to a_{ji} and $b_j(O_t)$ by the Expectation-Maximization [24] method. For this method, the HMM requires quite many recursive iterations as (2.57)-(2.62), which is not easily analyzed. The relation between the conventional Viterbi technique based on F-B HMM and the recursive Viterbi search based on k-means clustering is left for future research.

4.2.5 Appropriate Number of States

A clustering algorithm based on k-means helps to determine an effective number of states in a meaningful way [10]. Determining the number of states is also an important issue in designing a HMM [4, 16]. Generally, the number of states is determined roughly based on the expected number of identifiable acoustic phenomena in the utterance. For example, for a word model, five to ten states are often assigned to capture the phones in the utterances. For the phoneme model, three states are assigned for a HMM to model discrete phones. Since the recursive Viterbi search based on k-means clustering is configured to finding an state sequence which considers acoustical characteristics of a speech utterance, this technique can be exploited to automatically determine the appropriate number of states in the HMM.

One way of finding an appropriate number of states according to the sequence of

feature vectors of a speech utterance is found in [10]. As a basis for further development, we review this technique briefly.

There are a few ways of computing an appropriate number of clusters. Among them, the Davis-Bouldin (DB) index provides a relatively simple way of deciding an appropriate number of clusters [10]. It has been verified that the index does not depend on either the number of clusters nor the clustering method [10]. This index is as follows:

Given a partition of T objects into N clusters, one first defines the following measure of within-to-between cluster spread for all pairs of clusters (j, k) as

$$R_{j,k} = \frac{e_j + e_k}{m_{j,k}} \tag{4.33}$$

where e_j is the average error for the j^{th} cluster and $m_{j,k}$ is the Euclidean distance between the centers of the j_{th} and k_{th} clusters. Where the index for the k_{th} cluster is defined as

$$R_k = \max_{j \neq k}\{R_{j,k}\} \tag{4.34}$$

and the Davies-Bouldin index for the N-cluster clustering is defined as

$$DB(N) = \frac{1}{N}\sum_{k=1}^{N} R_k \quad \text{for } N > 1. \tag{4.35}$$

$DB(N)$ will be small for good clustering. The index is supposed to decrease monotonically as N decreases until the "correct" number of clusters is achieved for well-clustered data. The DB index is plotted against N and clustering is stopped when the index is apparently minimized.

The $DB(N)$ index can be easily implemented by the recursive Viterbi search based on k-means clustering because such a search technique employs the criterion

of minimizing the sum of Euclidean distances between the centroids of clusters and a sequence of feature vectors for a speech utterance. The distance information is the main element in DB computations. By using this method, let us find an appropriate number of states in our digit recognition problem.

$DB(N)$ versus N for the spoken utterances of "one, two, four," and "six" are displayed in Fig. 4.14 through 4.17 respectively. For word "two" and "four," we see that three states are adequate. In case of word "one" and "six," five states are suitable in the HMMs. We can apply this method to the other words. For the spoken digit recognition problem, therefore, five states will be appropriate. This result is consistent with the fact that conventionally we adopt four to six states for modeling a word in the HMM based on ML criterion.

Besides determining an adequate number of states, the $DB(N)$ index also let us know whether the objects are well-clustered or poorly-clustered. As explained previously, in the case of the well-clustered data, the $DB(N)$ index decrease monotonically as N increases until it reaches the "correct" number of states and then it increases. For arbitrary random data, such a trend does not occur [10]. For our example utterances, as shown in the figures, word "six" is better-clustered than the other utterances.

All the other spoken digits have similar patterns of $DB(N)$. In addition, the recursive Viterbi search based on k-means clustering is a meaningful technique for finding a meaningful state sequence of a speech utterance.

In the HMM, although the number of states of the HMM can be flexible, having an adequate number of states is significant for enhancing the performance of speech recognition system by tuning each model to more faithfully match the acoustic properties of its corresponding speech signals.

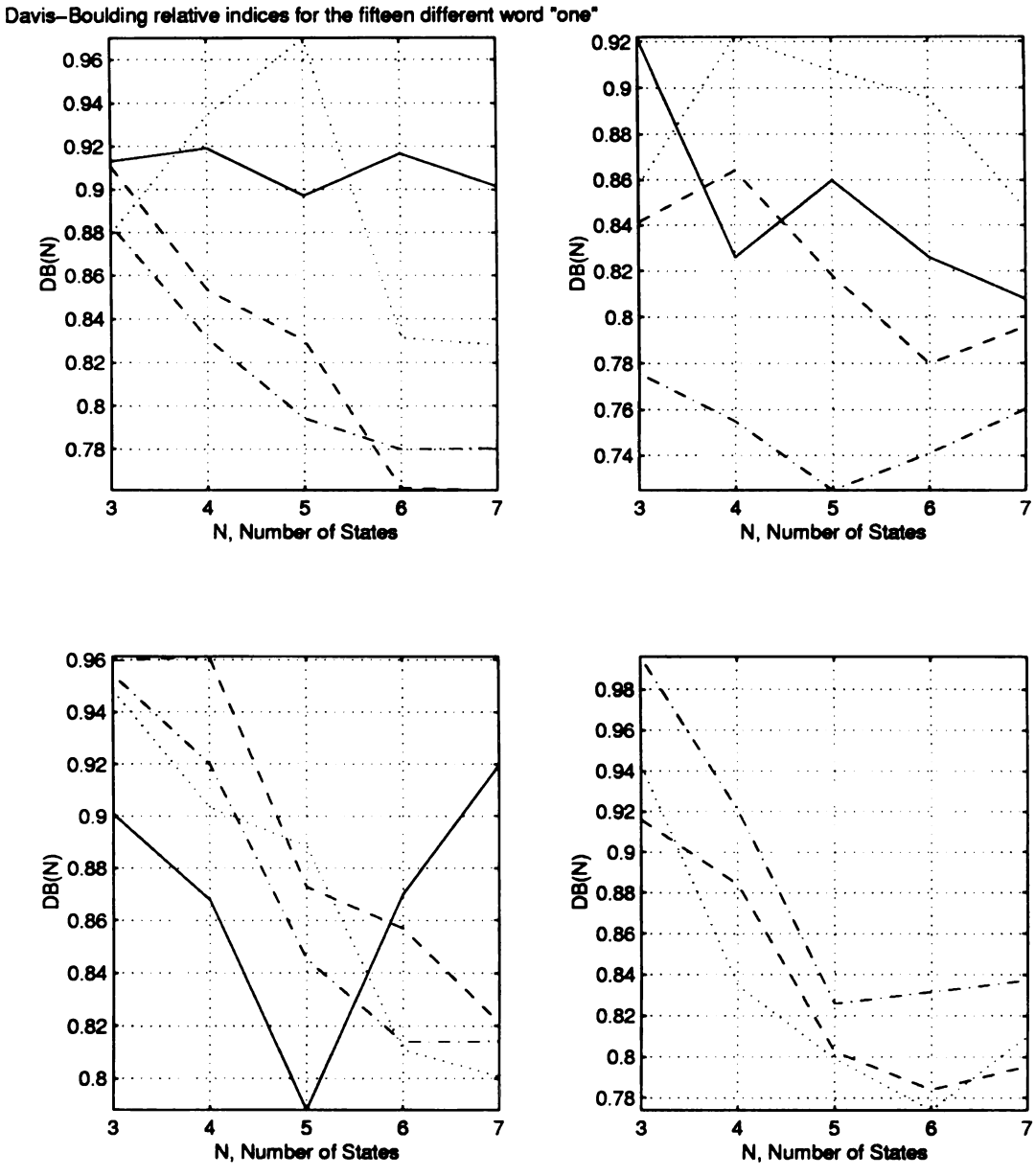

Figure 4.14: Davis-Bouldin relative index for fifteen utterances of spoken word "one."

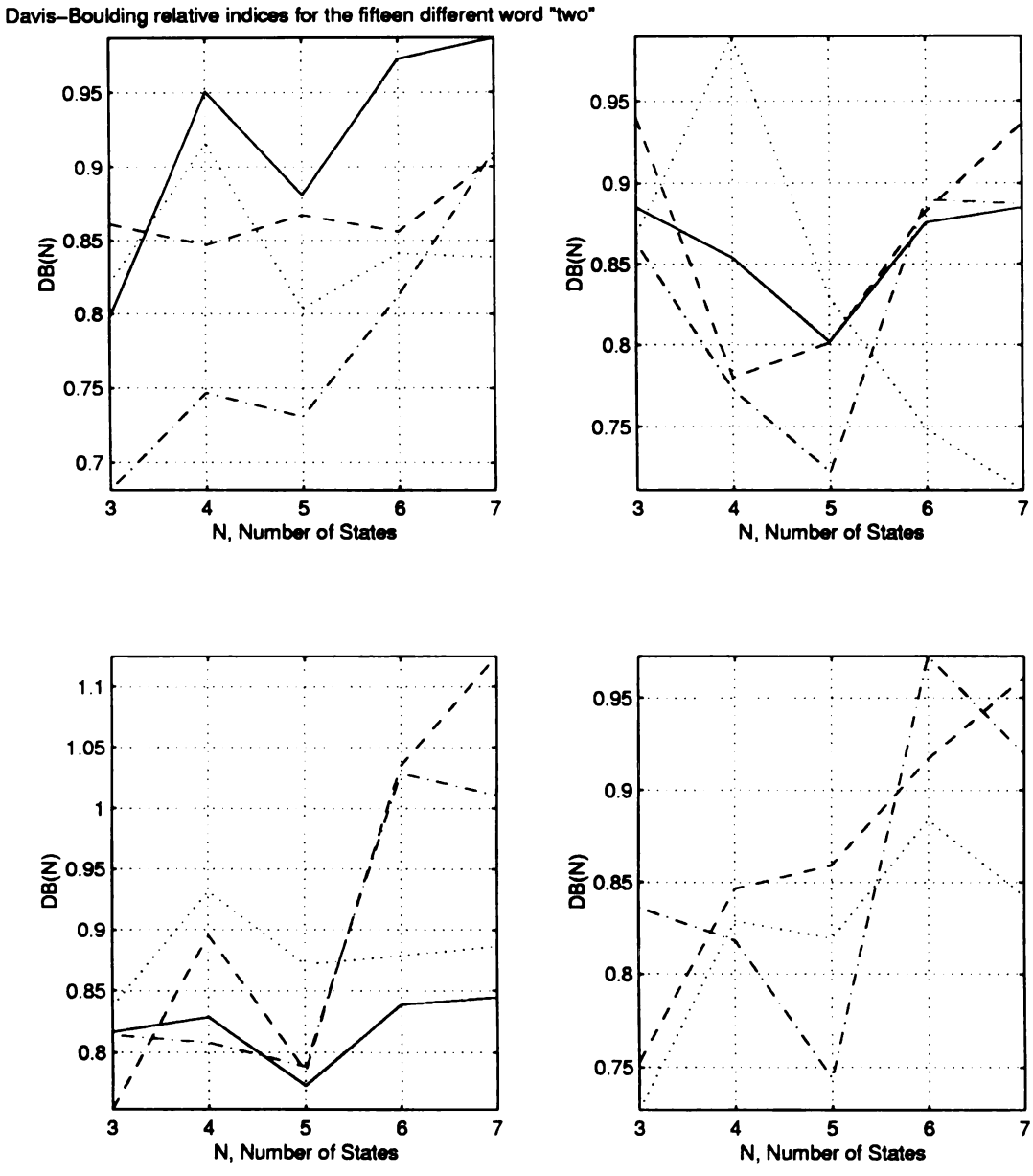

Figure 4.15: Davis-Bouldin relative index for fifteen utterances of spoken word "two."

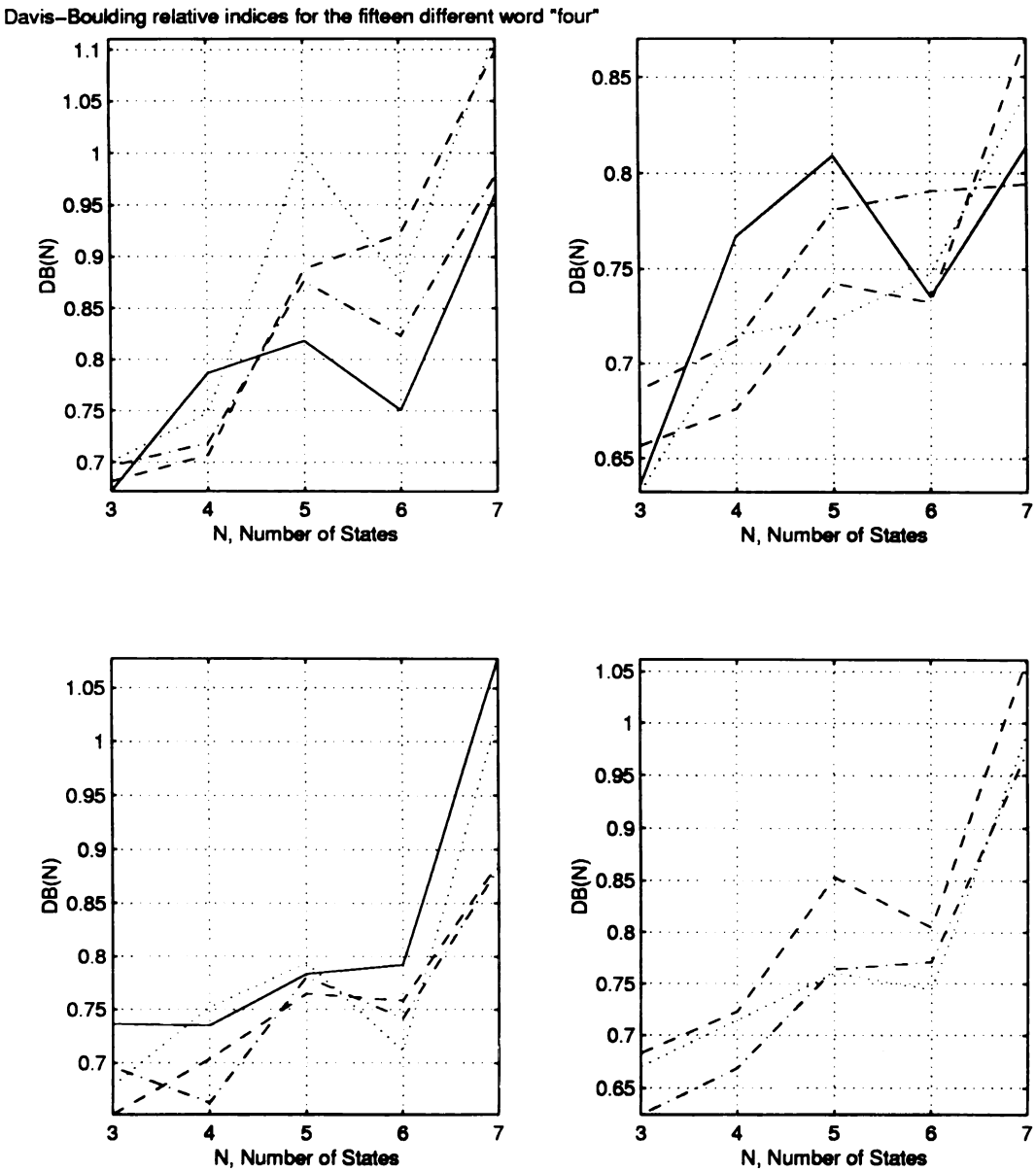

Figure 4.16: Davis-Bouldin relative index for fifteen utterances of spoken word "four."

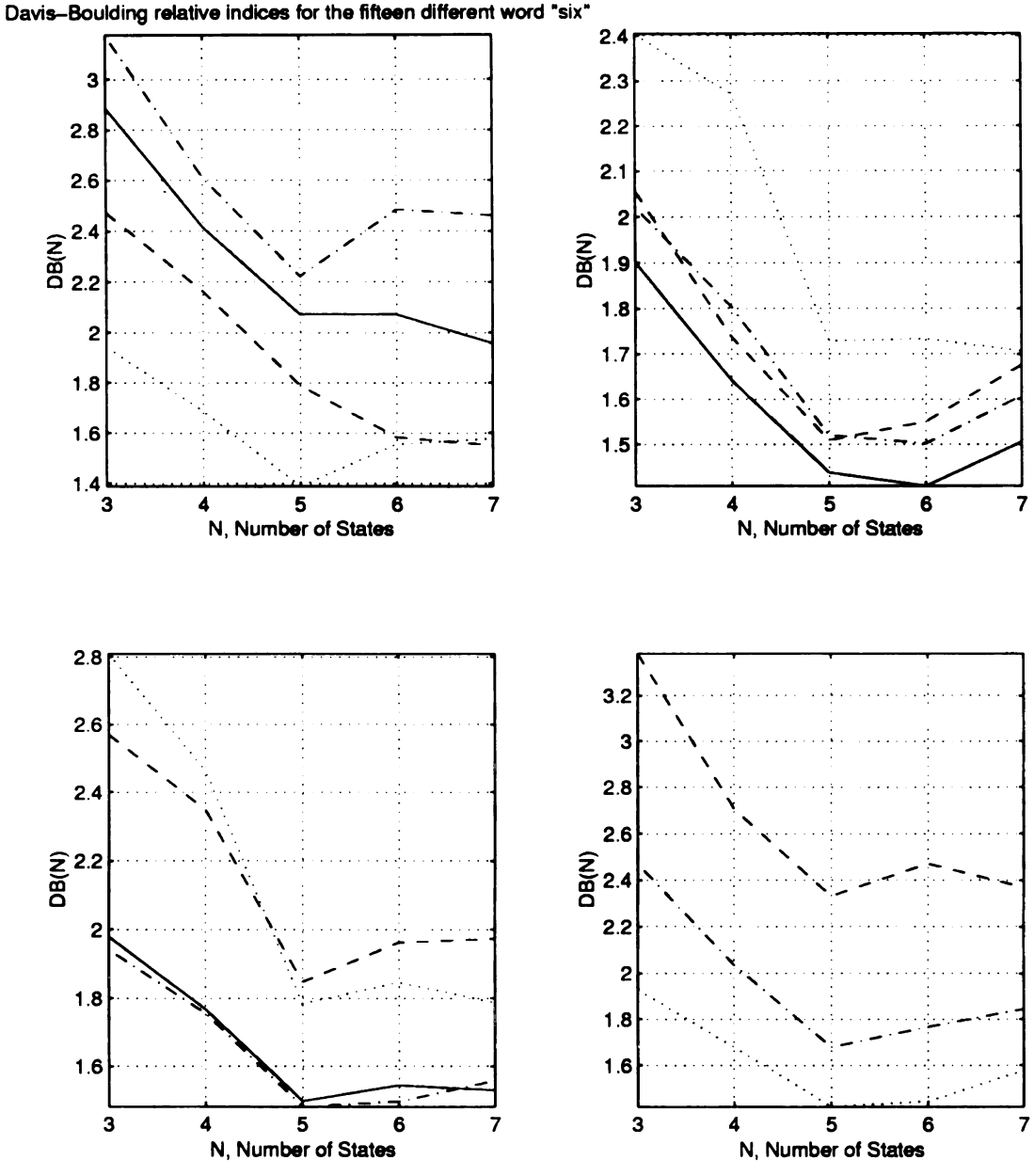

Figure 4.17: Davis-Bouldin relative index for fifteen utterances of spoken word "six."

4.2.6 Remark

First, let us examine the effect of initial clusters for the recursive Viterbi search based on k-means clustering. Different initial partitions can lead to different clustering results when the proposed algorithm is applied. This is because algorithms based on squared-error can inherently converge to local minima. This is especially true when the clusters are not well-separated. Therefore, in choosing the initial partitions, if possible, it is better to locate them far away from each other. Also one way to overcome the local minimum problem is to run the state searching algorithm with several different initial partitions. If they all lead to the same final partitions, we can be more confident that the global minimum of squared-error has been achieved.

Figures 4.18 and 4.19 are a set of several clustering results for word "four" and "six" with different initial clusters. For the word "four," we see that with the given initial clusters, there are two different patterns for the state sequence. For the word "six," we see more diverse patterns. Although the resulting state sequences are various depending on initial state partitions, they are still more consistent with the acoustical properties of a speech signal than those obtained by the Viterbi technique.

Also in this technique, it is possible to adjust the number of clusters by imposing a criterion on the state search algorithm so that it can be merged or split as in the ISODATA method [4, 10].

4.3 State Search by Set-Membership Identification

4.3.1 Original Thoughts about Exploiting the SM ID

In the HMM, no constraints are imposed on the speech signals (observations) at different time epochs except a nonstationary Markovian assumption. As another

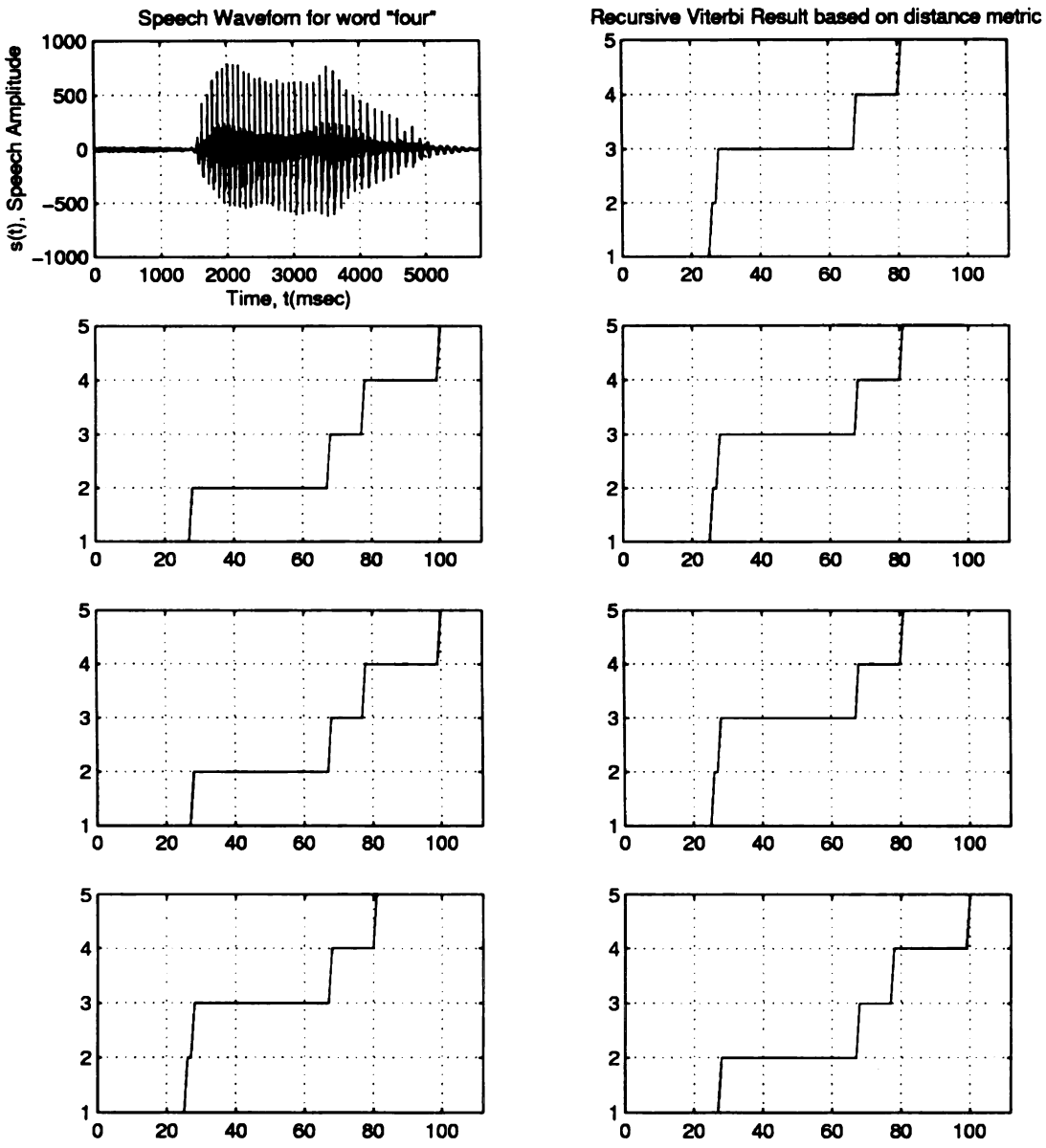

Figure 4.18: State search results for word "four" using the recursive Viterbi search based on k-means clustering with different initial clusters.

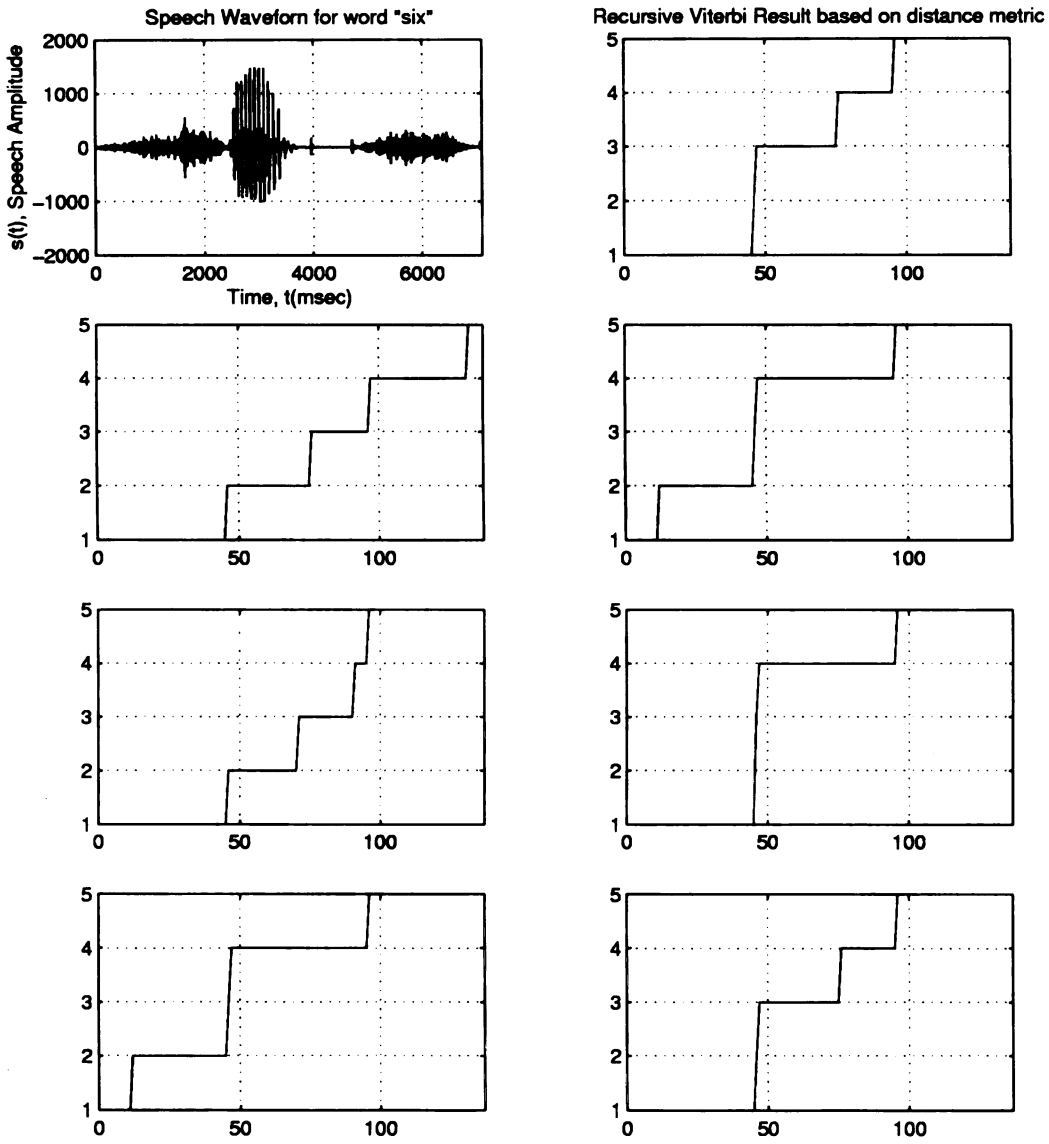

Figure 4.19: State search results for word "six" using the recursive Viterbi search based on k-means clustering with different initial clusters.

approach, for signals which cannot be modeled by stationary processes, for example speech, nonstationary autoregressive (AR) processes also have received considerable attention [77, 78]. In *hidden filter model* (HFM) approach, for example, O_t is not only conditioned by Markov transition probabilities, but also is dependent on $\{ O_{t-1}, \ldots, O_{t-p} \}$ for an appropriate p so that

$$O_t = \theta_i(1)O_{t-1} + \theta_i(2)O_{t-2} + \ldots + \theta_i(p)O_{t-p} + e_i(t). \tag{4.36}$$

Here $\theta_i^T = \left[\begin{array}{cccc} \theta_i(1) & \theta_i(2) & \cdots & \theta_i(p) \end{array} \right]$ is the vector of AR coefficients for state i and the driving sequence $e_i(t)$ is i.i.d Gaussian with mean μ_i and variance σ_i^2. The AR model parameters are made conditional on the state of the Markov chain i.

The original motive for exploiting the SM ID in speech recognition is taking a type of "filtering approach" which considers both stochastic and deterministic aspects of observations at the same time in speech recognition. The proposed model was thus called the *hidden filter for output probability distributions* (HF-OPD).

For the development of HF-OPD, the TIA HMM was employed. Applying z-transformation to (2.67) and (2.68), we have an input-output equation such as

$$\mathbf{y}(z) = \mathbf{B}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{u}(0)\delta(z). \tag{4.37}$$

For convenience, (4.37) has been transformed to

$$\mathbf{B}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{u}(0) = \mathbf{D}_L^{-1}(z)\mathbf{N}_L(z) \tag{4.38}$$

using matrix fraction description so that (4.37) can be represented as

$$\mathbf{D}_L(z)\mathbf{y}(z) = \mathbf{N}_L(z)\delta(z). \tag{4.39}$$

By inverse transformation, the corresponding temporal difference equation becomes

$$\mathbf{A_0 y}(t) + \ldots + \mathbf{A_N y}(t-N) = \mathbf{B_1}\delta(t-1) + \ldots + \mathbf{B_N}\delta(t-N) + e(t) \qquad (4.40)$$

with the added noise process. An estimated output probability distribution at each time was used as training data for this dynamics. To accommodate a more reasonable noise assumption, and to prevent excessive computation, a SM ID [81, 82] technique was proposed for this process.

However, it turned out that the SM ID approach using the TIA HMM is not easy. The main reason of the difficulty of applying the SM ID to the TIA HMM is the noise bound $e(t)$. Since the SM identification is based on a bounded-noise assumption, it is very important to have tenable noise bound. It is well known that noise bounds have significance influence on the performance of the SM identification [89, 90, 91, 92]. However, it was revealed that it is not easy to find an informative error bound when the training data for the SM ID is a probability distribution vector as $\mathbf{y}$ in (4.40). For example, if $\boldsymbol{x}(n)$ is a state vector and $\hat{\boldsymbol{x}}(n)$ is an estimate of $\boldsymbol{x}(n)$, and if

$$\boldsymbol{e}(n) = \boldsymbol{x}(n) - \hat{\boldsymbol{x}}(n), \qquad (4.41)$$

we have

$$\|\boldsymbol{e}(n)\|_2^2 \leq 2. \qquad (4.42)$$

Thus, the error bound of the input-output equation from the TIA HMM is too trivial and large. Similarly to this case, we had more difficulty in finding an informative error bound for $e(t)$ in a multivariable system as (4.40).

However, we found that the SM ID is still useful to identify a preliminary state sequence of speech signals because of its inherent characteristics of identifying the

dynamics of a model.

4.3.2 Background of the SM Identification

The SM identification was pioneered by Schweppe [81], Witsenhausen, and Bertsekas and Rhodes [82] in the late 1960s, in the domain of control and system science. In recent years, SM-based signal processing has been receiving considerable attention and has become increasingly popular around the world; especially for the "bounded error" (BE) problem which aims at characterizing the set of all parameter vectors consistent with prior bounds on the errors between the measurements and model outputs. These BE algorithms can be combined with various forms of least square error (LSE) signal processing algorithms with beneficial consequences [83, 84, 85, 87, 88, 89, 90, 91, 92].

Suppose that there is a general ARX (Auto-Regressive with Exogenous input) model,

$$y_n = \sum_{i=1}^{p} a_i y_{n-i} + \sum_{j=0}^{q} b_j u_{n-j} + v_n \tag{4.43}$$

$$= \theta^{*T} \mathbf{x}_n + v_n \tag{4.44}$$

where $\theta^{*T} = [a_1, \ldots, a_p, b_0, \ldots, b_q]$, $\mathbf{x}_n^T = [y_{n-1}, \ldots, y_{n-p}, u_n, u_{n-1}, \ldots, u_{n-q}]$, y_n and u_n are measurable outputs and inputs, respectively, and v_n is an unknown noise process. Let $m = p + q + 1$ and assume that for each time n, v_n is bounded by a magnitude, i.e.,

$$v_n^2 < \gamma_n \tag{4.45}$$

where $\{\gamma_n\}$ is a known positive sequence.

Let $w(n)$ be a parameter set at time n such that all $\theta \in w(n)$ are feasible parameter estimates of the model which are consistent with the error bounding in (4.45). In

conjunction with the model of form (4.44), $w(n)$, which is a "hyper-strip" region, can be expressed as

$$w(n) = \{\theta : \theta \in \mathbb{R}^m, (y_n - \theta^T \mathbf{x}_n)^2 \leq \gamma_n\}$$

which, when intersected over a given time range $t \in [1, n]$, usually form convex polytopes of feasible parameters

$$\Phi(n) = \bigcap_{i=1}^{n} w(i)$$

In general, $\Phi(n)$ is an irregular convex set and hence it is difficult to describe and track. But in conjunction with the WRLS (Weighted Recursive Least Square) processing, $\Phi(n)$ can be shown to be contained in a hyper-ellipsoid superset $\overline{\Phi}(n)$

$$\overline{\Phi}(n) = \{\theta : (\theta - \theta_n)^T \frac{\mathbf{C}(n)}{\kappa_n} (\theta - \theta_n) < 1\} \tag{4.46}$$

Where $\mathbf{C}(n)$ is the weighted covariance matrix,

$$\mathbf{C}(n) = \mathbf{C}(n-1) + \lambda_n \mathbf{x}_n \mathbf{x}_n^T \tag{4.47}$$

κ_n is a scalar quantity,

$$\kappa_n = \theta_n^T \mathbf{C}(n) \theta_n + \sum_{i=1}^{n} \lambda_n \gamma_n (1 - \frac{y_n^2}{\gamma_n}) \tag{4.48}$$

and θ_n, the center of $\overline{\Phi}(n)$, is the weighted LS estimate at time n using the weights $\{\lambda_i\}_{i=1}^n$. It can be shown that θ_n can be computed recursively using

$$\begin{aligned} \mathbf{C}(n) &\triangleq \mathbf{P}^{-1}(n) \\ G_n &= \mathbf{x}_n^T \mathbf{P}(n-1) \mathbf{x}_n \\ \varepsilon_n &= y_n - \theta_{n-1}^T \mathbf{x}_n \end{aligned} \tag{4.49}$$

$$
\begin{aligned}
\mathbf{P}(n) &= \mathbf{P}(n-1) - \frac{\lambda_n \mathbf{P}(n-1)\mathbf{x}_n \mathbf{x}_n^T \mathbf{P}(n-1)}{1+\lambda_n G_n} \\
\theta_n &= \theta_{n-1} + \lambda_n \mathbf{P}(n)\mathbf{x}_n \varepsilon_n \\
\kappa_n &= \kappa_{n-1} + \lambda_n \gamma_n - \frac{\lambda_n \varepsilon_n^2}{1+\lambda_n G_n}
\end{aligned}
$$

The SM-WRLS algorithm starts off with a large ellipsoid, $\overline{\Phi}(0)$, which contains all admissible values of the model parameter vector. The objective is to find the weight λ_n at time n so as to minimize the size of the membership set. Different criteria can be applied to the optimization process. One criterion is to minimize the volume ratio

$$
V(\lambda_n) = \frac{\det \mathbf{B}(n)}{\det \mathbf{B}(n-1)} \tag{4.50}
$$

where $\mathbf{B}(n) \triangleq \kappa_n \mathbf{C}^{-1}(n)$.

It has been shown[84, 88] that the optimal weight λ_n^* is the ***unique*** positive root of the quadratic equation

$$
\alpha_1 \lambda_n^2 + \alpha_2 \lambda_n + \alpha_3 = 0 \tag{4.51}
$$

where

$$
\alpha_1 = (m-1) G_n^2 \gamma_n \tag{4.52}
$$

$$
\alpha_2 = 2m G_n \gamma_n - \kappa_{n-1} G_n^2 - G_n \gamma_n + \varepsilon_n^2 G_n \tag{4.53}
$$

$$
\alpha_3 = m\gamma_n - m\varepsilon_n^2 - \kappa_{n-1} G_n \tag{4.54}
$$

4.3.3 State Search Using the SM Identification

The SM ID is useful not only for computing the unknown system parameters, but also for identifying the candidate state boundaries in a speech signal. In this section, a state search technique using the SM ID is discussed.

Concerned with a problem of finding a meaningful state sequence of a speech utterance, we focus on where the transitions of the parameter sets occurs rather than parameters of a model in the SM ID. The interesting part of the SM identification in connection with finding an appropriate sequence is that it takes advantage of volume or trace of an ellipsoid to describe the feasible values of parameters of a system. If the data are not informative, there is no change of volume or trace and this selective updating makes the SM theory useful. In addition, when the data come from a same source which may be regarded as one "state," the volume or trace will be monotonically non-increasing for new data. When the input data may come from a different source or "state," there exists a different set of parameters. Thus, it leads to a sudden change of the volume or trace (reset to the initial values) according to the theory of the SM ID. Therefore, if the volume or trace is tracked graphically or analytically with the SM algorithm, it is possible to get approximate information about plausible state boundaries in a speech utterance. In the SM, algebraically this implies that if the volume becomes negative at some points, those points could be considered where a significant change of characteristics of a signal happens.

The resulting state sequences of a spoken digit "six" deduced are shown in Fig. 4.20. The figure shows the sequences of three computed parameters, the volume of a bounding ellipsoid, κ_n, and informative points which indicates a sudden change of volume or κ_n in the time axis.

Here, the model is based on AR(3) which has three unknown parameters in a linear prediction model. However, the order can be changed. As described, the performance of the SM theory depends on the accurate bounds. In fact, knowing the bound information for the model is a difficult problem. There are a few of ways to estimate the noise bound. One method to approximately estimate the noise bound is

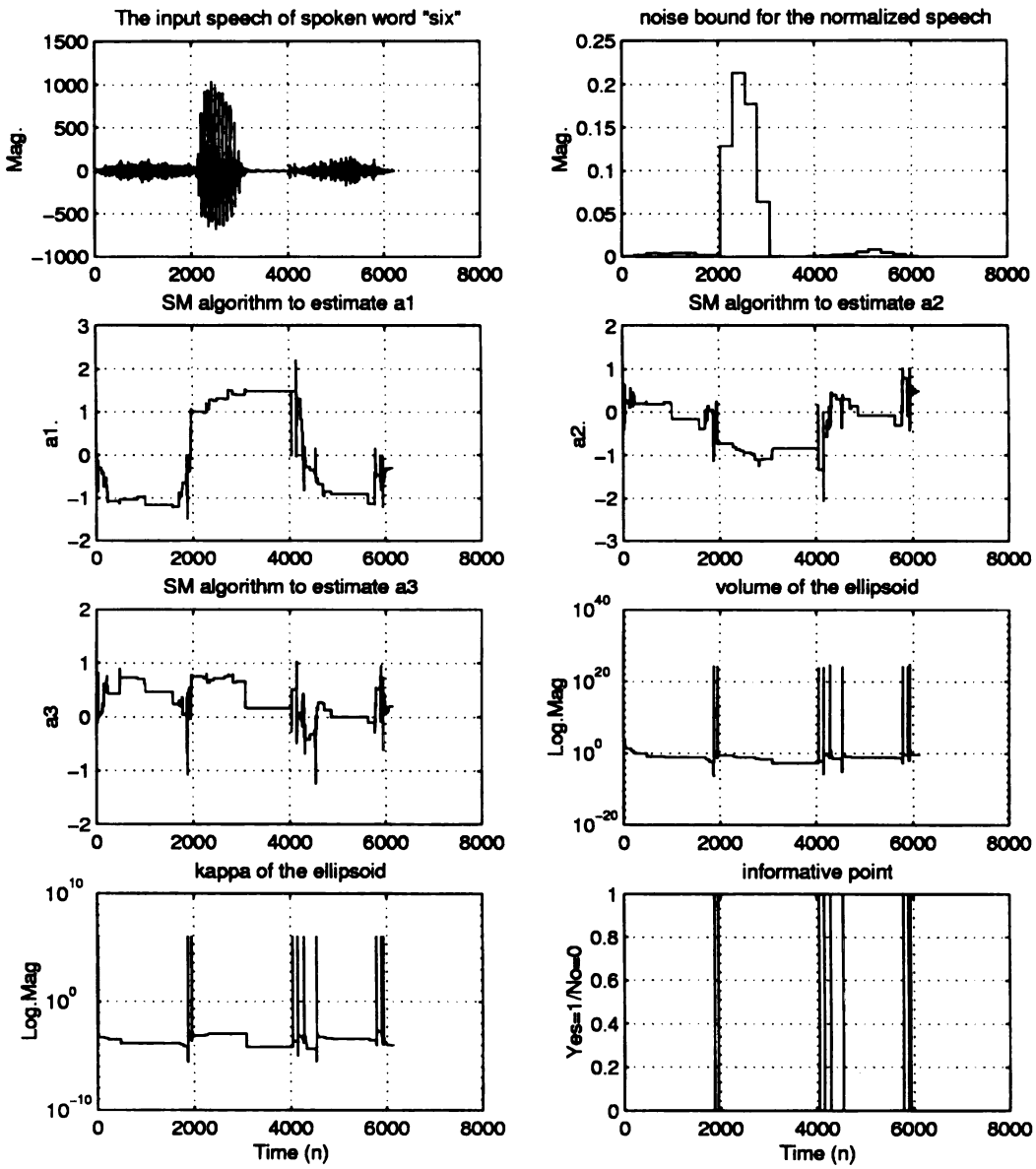

Figure 4.20: Some informative results regarding state segmentation by the SM technique for the word "six."

to use the short-term signal power of speech signals itself as

$$v^2(n) = \alpha \cdot \frac{\sum_{l=m-N+1}^{m} s^2(l)}{N} \tag{4.55}$$

where N is the frame size, α a constant, and

$$m = N \cdot \left(Q\left(\frac{n}{N}\right) + 1 \right). \tag{4.56}$$

Here, Q represents the integer part of quotient. These expressions imply that the noise is bounded uniformly within a N-size frame and that the noise is proportional to the average signal power of the frame. In addition, an amplitude or attenuation scalar α has been multiplied to adjust a noise bound depending on the application.

Conceptually, there is a transition of a set of parameters signifying the dynamics of a signal when the system changes the state. In Fig. 4.20, three parameters, a_1, a_2, and a_3 make a transition simultaneously at the points where physical status may change. More significantly, two major indicators, the volume of an ellipsoid covering the feasible parameter quantities and κ_n make sudden changes at those points so that the size of volume is reset to the initial size of volume. Those informative points indicate a change of the dynamics of a signal. Thus, by detecting the informative points, we had information about the transitions of acoustical phenomena of a speech signal.

The performance of this technique is heavily dependent on the accuracy of noise bound. Figure 4.21 and 4.22 are the results of state search for spoken digits "four" and "six" by the SM technique with various values of α. First, depending on the scale factor α, different sets of informative points are identified. Therefore, finding a good scale factor is significant for processing speech signals in this way. Second, most informative points are located in the boundary between unvoiced and voiced region as well as the unvoiced region. At the unvoiced region, the dynamics of AR model

changes rapidly. Thus, there occurs lots of informative points. For the transition region, it is as we expected. Therefore, the search technique is useful to detect a voiced/unvoiced regions of a signal by controlling a constant.

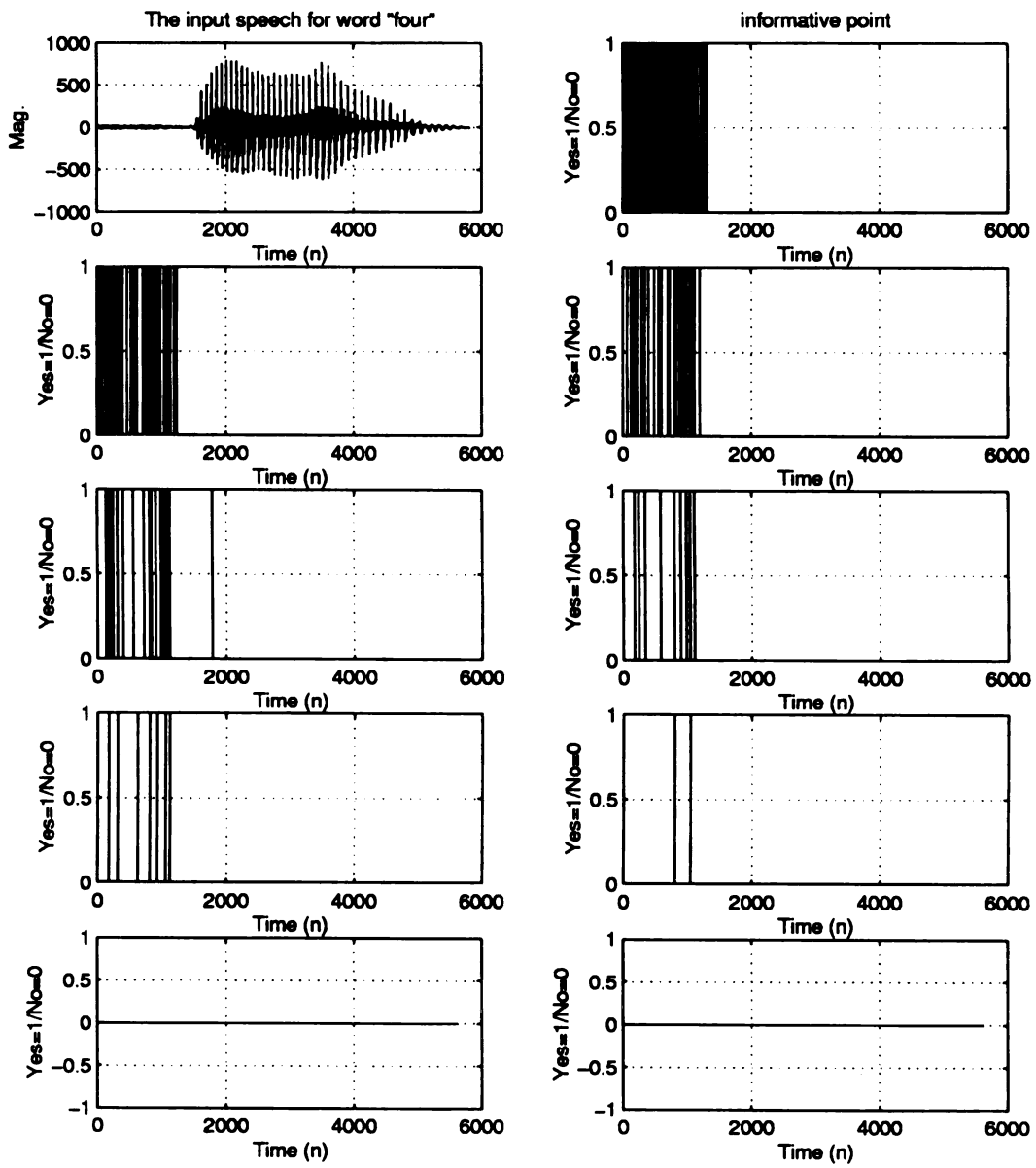

Figure 4.21: State segmentation result by the SM theory with various values of α for word "four."

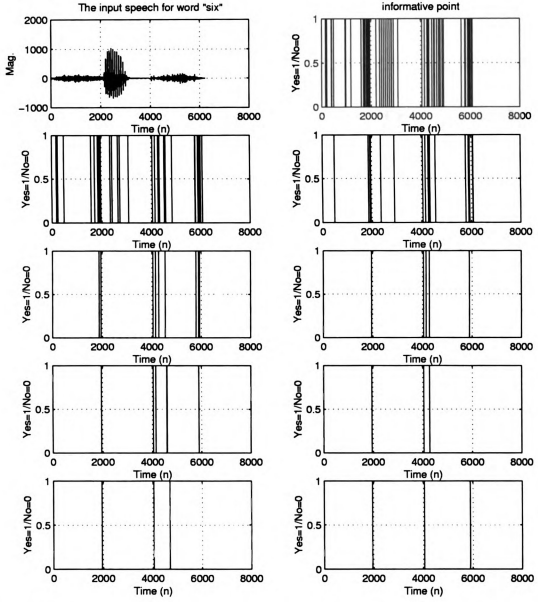

Figure 4.22: State segmentation result by the SM theory with various values of α for word "six."

Chapter 5

Conclusions and Future Research

5.1 Conclusions

This research is study about the HMM, a state-of-the-art technique in speech recognition. Most speech-based studies of the HMM are focused on the applications of the conventional F-B HMM.

First, the conventional F-B HMM has been transformed to the vector-matrix formulation. This general formulation admits diverse formulations of the three HMM problems and assist in the derivation of a more computationally efficient model.

Then, the main focus of this research is the reexamination of the TIA HMM, an approximate model of the conventional F-B HMM, with a vector-matrix formulation suggested by Snider *et al.* [7, 8]. In particular, we have focused on the inherent advantages with the TIA HMM in speech recognition. The TIA HMM reduces the storage requirements, improves computational efficiency, and increases numerical stability.

The TIA HMM has a natural vector-matrix formulation akin to a state-space model. This state-space formulation permits the reduction of the dimensions of the elements within a HMM population of tying some state variables so that the probability can be shared by the tied state variables. Therefore, by providing analytical

and empirical results to the TIA HMM, this research becomes the basis of viability of work of Snider [6] in related to the HMM compression.

In general speaking, this dissertation is an extension and more thorough examination of the work begun in [6] which falls short of reasonable explanations of applying the TIA HMM to speech recognition. In particular, this dissertation is focused on the analysis of the TIA HMM and its relation to the F-B HMM. By taking $\prod_{t=1}^{T} P(O_t)$ as likelihood measure for a speech utterance, the TIA HMM naturally causes to the generation of the extra likelihoods. However, we showed analytically and empirically that under some practical conditions such as a Bakis topology which has a diagonally dominant state-transition matrix, such an approximation is viable in some speech processing applications like spoken digit recognition.

Besides the analysis of vector-matrix formulation of the HMM undertaken here to demonstrate the viability of the TIA HMM, we derive some useful mathematical expressions for the F-B HMM as well as the TIA HMM. Such derived equations and accompanying results make it possible to re-exploit classic results of the HMM.

Next, we showed a couple of techniques to reconcile the F-B HMM to the TIA HMM although theoretically it is not possible to make the two models completely equivalent. However, these attempts are significant in showing similarities and relationships between some of the state variables of the models. Those approaches illuminate the operation of the HMMs while supporting the validity of the TIA HMM approximation of the F-B HMM.

It is concluded that the TIA HMM has some significant advantages resulting from loosening of the legal state path constraint which is implicitly required in the F-B HMM. However, we showed that such an illegal likelihood does not severely degrade the performance of spoken digit recognition with the TIA HMM.

Finally, we have proposed two new state search techniques. They are a new maximum likelihood approach (different from the conventional Viterbi technique which

also employs a ML criterion), and the acoustic distance approach. Comparing with the conventional Viterbi technique, by the new ML approach, we obtain an appropriate state sequence of a speech signal which is close to that of the Viterbi technique in a global sense that the global shapes of computed state sequences by the conventional Viterbi and the new ML approach are similar to each other. Also, as we obtain more computational efficiency in the TIA HMM, we also get the computational savings with this approach. Furthermore, the quite closeness of the state sequence between the conventional Viterbi approach and the new ML approach also becomes one viability condition of the TIA HMM.

The recursive Viterbi search based on k-means clustering is designed to examine the acoustic temporal variations of the speech signals and determine the corresponding state sequence. Thus, the resulting state sequence is more consistent with the acoustical evidence of a speech utterance.

In addition, the other possible approach of determining a possible state sequence is to use the SM ID. Even though the original motivation of adopting the SM ID in our research was to apply an adaptive linear filtering technique to find a parameters for $\mathcal{M}$ efficiently, we showed that SM ID technique is useful to find an appropriate state sequence of a speech signal. In particular, this SM technique is useful to detect boundaries of voiced, and unvoiced region of a speech signals. According to the application, we can choose a suitable state search method among proposed technique.

Recently, it is required for the speech recognition system to cover larger vocabularies as well as to support real-time speech applications. These applications require a robust and computationally fast processing of speech recognition system. Under certain environment of applications, however, one factor between the recognition rate and computational advantages is relatively more required than the other. In case that the computational aspect is relatively important without severe degradation of performance of the speech recognition system, the TIA HMM is a suitable technique

which allows to compromise the rate of recognition and resources with additional controllable factor according to the application.

The TIA HMM is one HMM approach which uses the state-space formulation. We conclude that the TIA HMM represents a viable alternative to the conventional F-B HMM in certain speech processing area such as digit recognition.

5.2 Future Research

Some of the open problems related to this research are summarized as follows:

- Using the vector-matrix formulation of the HMM, there are possibilities we can find some unknown characteristics of the F-B HMM which may be useful in speech-recognition technology albeit its inherent difficulty of time-varying characteristics.

- To develop and apply the TIA HMM more to the speech recognition problems, it is necessary to test and show the viability of the TIA HMM under the diverse domains of speech recognition problems. For a certain application, this technique may be favorable and for other applications, it may not. Therefore, it is significant to apply this technique to various application areas.

- In connection with the previous open problem, it is necessary to find or develop the conditions, if any, of the TIA HMM to improve the recognition performance without undermining the advantages of the TIA HMM.

- In this research, the application of the TIA HMM has been to the discrete HMM whose probabilities are all discrete components. However, application to continuous-density HMMs is also possible. In continuous observation HMM, $\mathcal{M}$ is to denote the elements of an HMM, namely N, $\{a_{ij}\}$, $\{b_j(\mathbf{x})\}$, and $\{\pi_i\}$. Here $b_j(\mathbf{x})$ is the density function of continuous observation process $\mathbf{x}$. Of course, the

same diagonalizing and compression idea can be applied to the state-transition matrix $\boldsymbol{A}$. However, for the continuous probability distribution, we need to find how to efficiently integrate the means and variances of the $b_j(\mathbf{x})$ and determine the effects on recognition performance.

- Since the HMM is a linear model, we can interpret the HMM in light of the adaptive signal processing technology. In the previous analysis, the SM ID technology has been exploited only to find a state sequence of a speech sequence. However, the SM ID theory is mostly applicable to the computation of unknown parameters of a model. More than anything else, the SM ID has an advantage in a selective updating of the parameters through the bounding ellipsoid based on the assumption of an known error bound. Therefore, the SM technique can be applied to deduce the state parameters in a different fashion from the F-B HMM. Furthermore, because the SM uses a bounding ellipsoid which includes all the feasible parameter values, the SM ID technique is assumed to make the speech recognition system robust. This is because using "set" of parameters instead of specific value for parameters, the SM ID theory can possibly assist to recover lost symbol information in the string during pronunciation or quantization.

- In connection with the application of the SM technique, we need to find an informative error bound in advance. Presently, the error bound of the input-output equation from the TIA HMM is too trivial and large for the development of HF-OPD. Also, the size of $\{\boldsymbol{A}, \boldsymbol{B}\}$ is still too large to process efficiently in the SM framework. As known, $\boldsymbol{B}$ is usually a sparse matrix which has many zeros in its elements and $\boldsymbol{A}$ is a triangular matrix. By considering matrix properties, a useful application of the SM ID technique to the HMM problems may emerge. This requires more intensive study.

Bibliography

[1] L.R. Rabiner, and B.H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine,* vol. 3, pp. 4-16, 1986.

[2] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE,* vol. 77, pp. 257-285, 1978.

[3] L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals,* Englewood-Cliffs, New Jersey: Prentice-Hall, 1978.

[4] J.R. Deller, Jr., J.H.L. Hansen, and J.G. Proakis, *Discrete Time Processing of Speech Signals,* (2nd ed.), New York: IEEE Press, 2000.

[5] J.G. Proakis, *Digital Communications,* (4th ed.), New York: McGraw-Hill, 2000.

[6] R.K. Snider, *Efficient Discrete Symbol Hidden Markov Model Evaluation Using Transformation and State Reduction* (M.S. Thesis), Michigan State University, East Lansing, 1990.

[7] J.R. Deller, Jr. and R. K. Snider, "Reducing redundant computation in HMM evaluation," *IEEE Trans. on Speech and Audio Processing,* vol. 1, Oct. 1993.

[8] J.R. Deller, Jr., and R.K. Snider, "'Quantized' hidden Markov models for efficient recognition of cerebral palsy speech," *Proc. 1990 IEEE Int. Symp. Circuits and Sys.*, New Orleans, vol. 3, pp. 2041-2044, May 1990.

[9] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach,* Englewood-Cliffs, New Jersey: Prentice-Hall, 1982.

[10] A.K. Jain, and R.C. Dubes, *Algorithms for Clustering Data,* Englewood Cliffs, New Jersey: Prentice-Hall, 1988.

[11] L.R. Rabiner, B.H Juang, and M.M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models with Continuous Mixture Densities," *AT & T Tech. J.,* vol. 64, pp. 1211-1234, July-Aug. 1985.

[12] M. Naito, L. Deng, and Y. Sagisaka, "Speaker clustering for speech recognition using the parameters characterizing vocal-tract dimensions," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Seattle, Washington, vol. 2, pp. 981-984, May 1998.

[13] A. Lazarides, Y. Normandin, and R. Kuhn, "Improving decision trees for acoustic modeling," *Proc. IEEE Int. Conf. Spoken Language,* Philadelphia, PA, vol. 2, pp. 1053-1056, Oct. 1996.

[14] C. Dugast, P. Beyerlein, and R. Haeb-Umbach, "Application of clustering techniques to mixture density modeling for continuous speech recognition," characterizing vocal-tract dimensions," *Proc.IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Detroit, vol. 1, pp. 524-527, May 1995.

[15] D.B. Paul, "Extensions to phone-state decision-tree clustering: Single tree and tagged clustering," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Munich, Germany, vol. 2, pp. 1487-1490, Apr. 1997.

[16] J. Zhang, Z. Hwang, and X. Wang, "Selection and analysis of HMM's state-number in speech recognition," *Proc. IEEE Int. Conf. Signal Processing,* Beijing, China, pp. 641-645, Oct. 1998.

[17] K.S. Fu, *Syntactic Pattern Recognition and Applications,* Englewood-Cliffs, New Jersey: Prentice-Hall, 1982.

[18] J.I. Tou and R.C. Gonzalez, *Pattern Recognition Principles,* Reading, Massachusetts: Addison-Wesley, 1974.

[19] C.-H. Lee, and K.S. Fu, "A stochastic syntax analysis procedure and its application to patter classification," *IEEE Trans. Computers,* vol 21, pp. 660-666, July 1972.

[20] J. Picone, "Continuous speech recognition using hidden Markov models," *IEEE ASSP Magazine,* vol. 7, pp. 26-41, July 1990.

[21] J.K. Baker, "The dragon system-An Overview," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 23, pp. 24-29, Feb. 1975.

[22] L.E. Baum, and J.A. Eagon, "An inequality with application 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.

[23] L.E. Baum, and T. Petrie, "Statistical inference for probabilistic function of finite state Markov chains," *Annals of Mathematical Statistics,* vol. 37, pp. 1554-1563, 1966.

[24] L.E. Baum, and T. Petrie, G. Soules *et al.*, "A maximization technique in the statistical analysis of probabilistic functions of Markov chains," *Annals of Mathematical Statistics,* vol. 41, pp. 164-171, 1970.

[25] C. Mitchell, M. Harper, and L. Jamieson," Comments on "Reducing computation in HMM evaluation"", *IEEE Trans. Speech and Audio Processing,* vol 2, Oct. 1994.

[26] C.T. Chen, *Introduction to Linear System Theory,* New York: Holt, Reinhart and Winston, 1970.

[27] W. Turin, "Unidirectional and parallel Baum-Welch algorithms," *IEEE Trans. Speech and Audio Processing,* vol. 6, Nov. 1998.

[28] C. Moler, and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Rev.,* vol. 20, pp. 801-836, 1978.

[29] R.L. Streit, "The moments of matched and mismatched hidden Markov models," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 38, Apr. 1990.

[30] H. Hjalmarsson, and B. Ninness, "Fast, non-iterative estimation of hidden Markov models," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Seattle, vol. 4, pp. 2253-2256, May 1998.

[31] M. Karan, B.D.O. Anderson and R.C. William, "An efficient calculation of the moments of matched and mismatched hidden Markov models," *IEEE Trans. Signal Processing,* vol. 43, Oct. 1995.

[32] R.J. Elliott, L. Aggoun, and J.B. Moore, *Hidden Markov models : Estimation and Control,* New York: Springer-Verlag, 1995.

[33] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition,* Englewood Cliffs, New Jersey: Prentice-Hall, 1993.

[34] D. O'Shaughnessy, *Speech Communication: Human and Machine,* Reading, Massachusetts: Addison-Wesley, 1987.

[35] J. Allen, "How do humans process and recognize speech?," *IEEE Trans. Speech and Audio Processing,* vol. 2, Oct. 1994.

[36] R.A. Horn, and C.R. Johnson, *Topics in Matrix Analysis,* New York: Cambridge University Press, 1991.

[37] J.M. Ortega, *Matrix Theory: A Second Course,* New York: Plenum Press, 1987.

[38] B. Noble, and J. W. Daniel, *Applied Linear Algebra,* Englewood Cliffs, New Jersey: Prentice-Hall, 1988.

[39] G.H. Golub, and C.F. Van Loan, *Matrix Computations,* (2nd ed.), Baltimore: Johns Hopkins Press, 1989.

[40] X.D. Huang, Y. Ariki and M.A. Jack, *Hidden Markov Models for Speech Recognition,* Edinburgh: Edinburgh University Press, 1990.

[41] S.I. Resnick, *Adventures in Stochastic Processes,* Boston: Birkhauser, 1992.

[42] M.J.R. Healy, *Matrices for Statistics,* , New York: Oxford University Press, 1986.

[43] S.R. Searle, *Matrix Algebra Useful for Statistics,* New York: Wiley, 1982.

[44] H. Minc, *Nonnegative Matrices,* New York: John Wiley & Sons, 1988.

[45] A. Graham, *Nonnegative Matrices and Applicable Topics in Linear Algebra,* New York: John Wiley & Sons, 1987.

[46] R.S. Varga, *Matrix Iterative Analyses,* Englewood Cliffs, New Jersey: Prentice-Hall, 1962.

[47] D.K. Faddeev, *Computational Methods of Linear Algebra,* San Francisco: W. H. Freeman and Company, 1963.

[48] A. Jennings, *Matrix Computation for Engineers and Scientists,* New York: John Wiley & Sons, 1977.

[49] M. Aoki, *State Space Modeling of Time Series,* New York: Springer-Verlag, 1987.

[50] W.H.M. Zijm, *Nonnegative Matrices in Dynamic Programming,* Amsterdam: Mathematisch Centrum, 1983.

[51] E. Seneta, *Non-negative Matrices and Markov Chains,* (2nd ed.), New York: Springer-Verlag, 1981.

[52] E. Bodewig, *Matrix Calculus,* Amsterdam: North-Holland, 1959.

[53] S.S Haykin, *Adaptive Filter Theory,* Englewood Cliffs, New Jersey: Prentice-Hall, 1996.

[54] J.S. Lim and A.V. Oppenheim (editors), *Advanced Topics in Signal Processing,* Englewood Cliffs, New Jersey: Prentice-Hall, 1988

[55] A.V. Oppenheim, and R.W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

[56] C.H. Lee and L.R. Rabiner, "A frame synchronous network search algorithm for connected word recognition", *IEEE Trans. Acoustics, Speech and Signal Processing,* vol. 37, pp. 1649-1658, Nov. 1989.

[57] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Information Theory,* vol. 13, pp. 260-269, Apr. 1967.

[58] A.J. Viterbi, and J.K. Omura, *Principles of Digital Communications and Coding,* New York: McGraw-Hill, 1979.

[59] F. Jelinek, "A fast sequential decoding algorithm using a stack", *IBM J. Research and Development,* vol. 13, pp. 675-685, Nov. 1969.

[60] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE,* vol. 73, Nov. 1985.

[61] M. Ostendorf, V.V. Digalakis, and O.A. Kimball, "From HMM's to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans. Speech and Audio Processing,* vol. 4, Sept. 1996.

[62] Sing-Tze Bow, *Pattern Recognition: Applications to Large Data-Set Problems,* New York: M. Dekker, 1984.

[63] J. Dai, I.G. Mackenzie, and E.M. Tyler, "Stochastic modeling of temporal information in speech for hidden Markov models," *IEEE Trans. Speech and Audio Processing,* vol. 2, Jan. 1994.

[64] N. Nocerino, F.K. Soong, L.R. Rabiner, and D.H. Klatt, "Comparative study of several distortion measures for speech recognition," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Tampa, vol. 1, pp. 25-28, 1985.

[65] L. Rabiner, J.G. Wilpon, and F.K. Soong, "High performance connected digit recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 36, Aug. 1989.

[66] P. McGoldrick, "Did you hear what I said?," *Electronic Design,* Oct. 1996.

[67] A.B. Poritz, "Hidden Markov models: A guided tour," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* vol. 1, pp. 7-13, 1988.

[68] P. Kenny, M. Lennig, and P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition," *IEEE Trans. Speech and Audio Processing,* vol. 38, pp. 220-225, Feb. 1990.

[69] A.B. Poritz, "Linear predictive hidden Markov models and the speech signal," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* vol. pp. 1291-1294, May 1982.

[70] B. Juang, and L.R Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 33, pp. 1404-1413, Dec. 1983.

[71] H. Sheikhzadeh, and L. Deng, "Waveform-based speech recognition using hidden filter models: parameter selection and sensitivity to power normalization," *IEEE Trans. Speech and Audio Processing,* vol. 2, pp. 80-89, Jan. 1994.

[72] L. Deng, "A stochastic model of speech incorporating hierarchical nonstationarity," *IEEE Trans. Speech and Audio Processing,* vol. 4, pp. 471-474, Oct. 1993.

[73] B. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains, " *IEEE Trans. Information Theory,* vol. 32, pp. 307-309, Mar. 1986.

[74] N. Tishby, "On the application of mixture AR hidden Markov models to text independent speaker recognition," *IEEE Trans. Signal Processing,* vol. 39, pp. 563-570, Mar. 1991.

[75] L. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources, " *IEEE Trans. Information Theory,* vol. 28, pp. 729-734, Sep. 1982.

[76] B. Atal, and S. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave, " *J. Acoust. Soc. Amer.,* vol. 50, no. 2, pt. 2, pp. 637-655, 1971

[77] L. Liporace, "Linear estimation of nonstationary signals," *J. Acoust. Soc. Amer.,* vol. 58, pp. 1288-1295, 1975.

[78] Y. Grenier, "Time-dependent ARMA modeling of nonstationary signals," *IEEE Trans. Speech and Audio Processing,* vol. 31, pp. 899-911, 1983.

[79] R. Bakis, "Continuous speech word recognition via centisecond acoustic states," *Proc. 91st Annual Meeting of the Acoustic Society of America,* Washington, D.C., 1976

[80] J. J. Ford, and J. B. Moore, "On adaptive HMM state estimation," *IEEE Trans. Signal Processing,* vol. 46, Feb. 1998.

[81] F.C. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Trans. Automatic Control,* vol. AC-13, pp. 22-28, Feb. 1968.

[82] D.P. Bertsekas, and I.B. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Trans. Automatic Control,* vol. AC-16, pp. 117-128, Apr. 1971.

[83] E. Fogel, "System identification via membership set constraints with energy constrained noise," *IEEE Trans. Automatic Control,* vol. AC-24, pp. 752-758, Oct. 1979.

[84] E. Fogel, and Y.F. Huang, "On the value of information in system identification - bounded noise case," *Automatica,* vol. 18, pp. 229-238, 1982.

[85] S. Dasgupta, and Y.F. Huang, "Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise," *IEEE Trans. on Information Theory,* vol. IT-33, pp. 383-392, May 1987.

[86] J.P. Norton, "Recursive computation of inner bounds for the parameters of linear model," *Int. J. Control,* vol. 50, pp. 2423-2430, 1989.

[87] J.R. Deller, Jr., and S.F. Odeh, "Implementing the optimal bounding ellipsoid algorithm on a fast processor," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Glasgow, vol. 2, pp. 1067-1070, May 1989.

[88] J.R. Deller, Jr., "Set-membership identification in digital signal processing," *IEEE ASSP Magazine,* vol. 6, pp. 4-22, Oct. 1989.

[89] J.R. Deller, Jr., and T.C. Luk, "Linear prediction analysis of speech based on set-membership theory," *Computer Speech and Language,* vol. 3, pp. 301-327, Oct. 1989.

[90] J.R. Deller, M. Nayeri, and M.S. Liu, "Unifying the landmark developments in optimal bounding ellipsoid identification," *Int. J. on Automatic Control and Signal Processing,* vol. 8, pp. 43-60, Jan-Feb. 1994.

[91] M. Nayeri, M.S. Liu, and J.R. Deller, "An interpretable and converging set-membership algorithm," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Minneapolis, vol. 4, pp. 472-475, Apr. 1993.

[92] M. Nayeri, J.R. Deller, and M.S. Liu, "A converging optimal ellipsoid algorithm with volume minimization," *Proc. 26th Annual Asilomar Conf. Signals, Systems, and Computers,* Monterey, pp. 20-24, Oct. 1992.

[93] A.K. Rao, Y.F. Huang, and S. Dasgupta, "ARMA parameter estimation using a novel recursive estimation algorithm with selective updating," *IEEE Trans. Speech and Audio Processing,* vol. 38, pp. 447-457, Mar. 1990.

[94] T. Kailath, *Linear Systems,* Englewood-Cliffs, New Jersey: Prentice-Hall, 1980.

[95] K.F. Lee, *Automatic Speech Recognition, the Development of the SPHINX System,* Boston: Kluwer, 1989.

[96] T. Soderstrom, and P. Stoica, *System Identification,* Englewood-Cliffs, New Jersey: Prentice-Hall, 1989.