Autoregressive Hidden Markov Model with Application in an El Niño Study

A Thesis Submitted to the College of
Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Mathematics and Statistics
University of Saskatchewan
Saskatoon

by

Tang Xuan

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mathematics and Statistics

University of Saskatchewan

Saskatoon, Saskatchewan, Canada

S7N5E6

ABSTRACT

Hidden Markov models are extensions of Markov models where each observation is the result of a stochastic process in one of several unobserved states. Though favored by many scientists because of its unique and applicable mathematical structure, its independence assumption between the consecutive observations hampered further application. Autoregressive hidden Markov model is a combination of autoregressive time series and hidden Markov chains. Observations are generated by a few autoregressive time series while the switches between each autoregressive time series are controlled by a hidden Markov chain. In this thesis, we present the basic concepts, theory and associated approaches and algorithms for hidden Markov models, time series and autoregressive hidden Markov models. We have also built a bivariate autoregressive hidden Markov model on the temperature data from the Pacific Ocean to understand the mechanism of El Niño. The parameters and the state path of the model are estimated through the Segmental K-mean algorithm and the state estimations of the autoregressive hidden Markov model have been compared with the estimations from a conventional hidden Markov model. Overall, the results confirm the strength of the autoregressive hidden Markov models in the El Niño study and the research sets an example of ARHMM's application in the meteorology.

ACKNOWLEDGMENTS

I would like to express my thanks to my supervisor Professor W.H.Laverty for his guidance. At the same time, I would also like to express my thanks to all members of my advisory committee for their reading this thesis.

Last but not least, I want to thank my family and friends, for their moral support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1
HIDDEN MARKOV MODELS

1.1 General Introduction

El Niño is a disruption of the ocean-atmosphere system in the tropical Pacific. It is characterized by a large scale weakening of the trade winds and the warming of the sea surface in the eastern and central equatorial Pacific ocean. It was initially recognized by fishermen in the South America when they observed the unusual warming in the Pacific ocean. Because the phenomenon tends to arrive around Christmas, it gains the name "El Niño" which means "The Little Boy" in Spanish.

El Niños have important consequences for weather around the globe. Not only have they caused great reductions in marine fish and plant life along the east Pacific coast in several years, but also they were responsible for many destructive flooding and drought in the West Pacific which lead to the displacement of thousands from their homes. According to the meteorologic records, El Niños occur irregularly at intervals of 2-7 years, with an average of 3-4 years. During the past forty years, there have been about ten major El Niño events recorded. Among those, the worst one occured in 1997. The sea surface temperature for September 1997 was the highest in the last 50 years. Furthermore, in late September easterly winds over the equatorial Pacific between 150E and 120W decreased the most in the last 30 years.

There is no doubt of the existence of El Niños. As a physical occurrence it is just

as real as rainfalls or thunderstorms. But the way it works has many theories. In this thesis, we assume that the ocean-atmosphere system of Pacific Ocean has two (or more) distinct states, *normal state* and *abnormal state*(or *El Niño state*). An El Niño is the result of a switch from the normal state to the abnormal state. The switches between normal state and abnormal state are unseen, but can be inferred from the numerical observations such as the sea surface temperatures and trade wind intensities. Furthermore, we assume that the chronological state sequence follows a Markov process. In this way, we could utilize a sophisticated mathematical model, autoregressive hidden Markov model (ARHMM), in the research of El Niño.

Autoregressive hidden Markov model is a natural combination of hidden Markov model and autoregressive time series model. Following this introduction is an introduction of the basic theories of Hidden Markov Models. In Chapter Two we will present a general introduction of time series models, followed by the definition and estimation of an advanced model, autoregressive hidden Markov model in Chapter Three. Finally, tests and an application of ARHMM in the El Nino are performed and related results are discussed in the Chapter Four.

1.2 Introduction of Hidden Markov Models

Imagine a coin-tossing game in which two coins are alternatively tossed in a sequence. The choice of a coin and the switches between the two coins are behind

the scenes. What is observed is the outcomes of the tossing: a sequence of heads or tails (e.g. THHHTTHTHHT...) which will be called *observation sequence* or simply *observations* or *observation data*. To appreciate how the observation sequence are influenced by the bias and the order of coin-tossing, suppose you know coin #1 has much higher bias to produce a tail than coin #2, which is assumed to be a fair coin. We also assume that in every turn the two coins are equally likely to be chosen, then it is natural to expect there will be more tails than heads in the whole sequence, especially when the observation sequence is fairly long. In turn, though you don't know anything about the bias or choices of the coins, when there are much more tails appearing, you would suspect one of or both the coins are tail-biased. Actually, this simple coin-tossing game characterize a class of probabilistic models which is called *Hidden Markov Model*. In hidden Markov model, each observation is partially decided by its current state (the current choice of coins). Since the state sequence is unseen, we call it "hidden". The state sequence is assumed to follow a Markov process in which the current state depends only on its latest previous state probabilistically. In most applications where hidden Markov models are used, one would have to draw a probabilistic inference about the hidden states based on the observation data.

The basic concept and theories of hidden Markov models were introduced by Baum and his colleagues in late 1960's. Then in the following a couple of years the main interests of research remains purely in its mathematical structure and properties,

probably because the inference of hidden Markov models would involve huge amounts of computations which cannot be handled at that time. Until early 1980's, when the prevalence of electronic computer greatly facilitates the mathematical computation in the whole scientific world, Rabiner, Juang and their colleagues have published a series papers ([14]-[18]) on speech recognition based on the hidden Markov models. In their models, every time when a word is vocalized, it essentially goes through a series states. The sound is relatively smooth when staying in the same states and will fluctuate when undergoing frequent state changes. Since everybody has their unique pattern of pronunciation, it is possible to identify a man's voices through recognizing his particular hidden Markov model. Due to their rich and practical mathematical structure, HMMs have become more and more popular in various important applications. For example, many recent works in economics are based on J.Hamilton's [10] time series model with changes in regime which is essentially a class of hidden Markov models. In many of them, the fluctuating economic numbers such as stock index and GDP are very much influenced by the business cycle which can be seen as the hidden states with seasonal changes. Through the recognition and estimation of hidden states, they could better understand the mechanism of business cycle. Another booming application of HMMs is in the bio-technology where people use a particular class of hidden Markov model (profile HMM) to model proteins and DNA strings, recognize genes and so on. Other applications of hidden Markov model includes optical

characters recognitions, natural language understanding and climatological forecasts, etc.

This introduction is followed by the formal definition and the most basic problems of HMM.

1.3 Definition of Hidden Markov Models

The coin-tossing example in the last section gives us an intuitive idea of what a hidden Markov model is. Now we will formally define the model.

A HMM is characterized by the following elements:

1. N, the number of states in the model. In the coin tossing example, the states correspond to the choice of the coins (i.e. two possible states). We will denote the state at time t as X_t throughout the thesis.

2. M, the number of distinct observation symbols in each states, namely the alphabet size. For the coin tossing example, the observation symbols are simply the "head" and the "tail". We will use Y_t to denote the observation symbol at time t.

3. T, The length of the observation sequence. So the states sequence can be written as $\{X_1, X_2, \ldots, X_T\}$ and the observations sequence would be $\{Y_1, Y_2, \ldots, Y_T\}$.

4. A set of transition probability $A = \{a_{ij}\}$, where

$a_{ij} = P[X_{t+1} = j | X_t = i], \qquad 1 \leq i, j \leq N.$

Note $\{a_{ij}\}$ subjects to the probability constraints:

$a_{ij} \geq 0 \qquad$ for all $1 \leq i, j \leq N$, and

$\sum_{j=1}^{N} a_{ij} = 1 \qquad$ for all $1 \leq i \leq N$.

5. The observation symbol probability (also called emission probability) distribution in state $i : B = \{b_i(m)\}$,

   $b_i(m) = P(v_m \; at \; time \; t | X_t = i)$,

   where $1 \leq i \leq N$, $1 \leq m \leq M$ and v_m is the m^{th} symbol in the observation alphabet.

   When the emission probability distribution is continuous, we denote

   $b_i(y) = f(y|\theta_i)$ the conditional probability distribution of Y_t given $X_t = i$, where θ_i is unknown parameter(s) of the distribution in state i. In the most common case when the distribution is normal, $\theta_i = (\mu_i, \Sigma_i)$, where μ_i and Σ_i stand for the mean and covariance matrix in state i, respectively.

6. The initial state distribution $\pi = \{\pi_i\}$,

   $\pi_i = P[X_1 = i], \qquad 1 \leq i \leq N.$

From the definitions above, it is clear that a complete specification of a HMM involves three model parameters (N,M and T)and three sets of probability parameters

(A, B and π). For convenience, we use a compact notation $\lambda = (A, B, \pi)$ to represent the complete set of parameters of the model throughout the thesis.

1.4 Three Basic Problems and Two Assumptions

To use the hidden Markov model to the real-world application, there are three very fundamental problems need to be solved:

1. Given the HMM $\lambda = (A, B, \pi)$, What is the probability of generating a specific observation sequence $\mathbf{Y}$=$\{Y_1, Y_2, \ldots, Y_T\}$? i.e. How to compute $P(\mathbf{Y}|\lambda)$?

2. Given the observation sequence $\mathbf{Y}$=$\{Y_1, Y_2, \ldots, Y_T\}$, how to determines the states sequence $\mathbf{X}$=$\{X_1, X_2, \ldots, X_T\}$?

3. Given the observation sequence $\mathbf{Y}$=$\{Y_1, Y_2, \ldots, Y_T\}$, how to estimate the parameters $\lambda = (A, B, \pi)$ of the HMM?

Throughout the whole thesis,"*observation probability*", $P(\mathbf{Y}|\lambda)$, denotes the probability or likelihood of the occurrence of the observation sequence $\mathbf{Y}$ given the parameter set λ. Please note λ is not a random variable hence $P(\cdot|\lambda)$ may not be regarded as a conditional probability. For discrete distribution, a more accurate expression might be $P(y = \mathbf{Y}; \lambda)$, the probability of a random variable y equals to the observation sequence $\mathbf{Y}$ given the parameter set λ . When the distribution of observation

variable y is continuous, $P(\mathbf{Y}|\lambda)$ can be seen as a "probability function" of λ which is algebraically equal to the likelihood function $L(\lambda|\mathbf{Y})$. This succinct notation of probability, instead of the corresponding likelihood function, has been adopted by the major literatures of HMM to facilitate the usage of probability theorems. We will follow this notation throughout the thesis.

To ensure the tractability of these problems, we have to make two assumptions for the structure of HMM:

1. **Markov Assumption** : At any time t, the probability of generating the next state depends only on the current state. i.e.

$$P(X_{t+1}|X_t, X_{t-1}, \cdots, X_0) = P(X_{t+1}|X_t) \tag{1.1}$$

for all t.

2. **Independency Assumption** : The probability distribution of generating current observation symbol depends only on the current state. This assumption indicates

$$P(\mathbf{Y}|\mathbf{X}, \lambda) = \prod_{t=1}^{T} P(Y_t|X_t, \lambda), \tag{1.2}$$

in which $\mathbf{Y} = \{Y_1, Y_2, \cdots, Y_T\}$ and $\mathbf{X} = \{X_1, X_2, \cdots, X_T\}$ denote the observation sequence and state sequence , respectively.

The Markov assumption of HMM will be engaged throughout the thesis. But the independency assumption will be applied in this chapter only. In Chapter Three, we will introduce autoregressive hidden Markov Models in which the current observations do not only depend on the current state, but also on the past observations.

Then in the next few sections, we will discuss the solutions for this three basic questions.

1.5 Solving Problem One – Forward-Backward Method

$P(\mathbf{Y}|\lambda)$, the observation probability given the model $\lambda = (A, B, \pi)$, can also be seen as a measure of how well the given observation sequence $\mathbf{Y}$ fitting into the model. With this measure, it allows us to choose the best HMM amongst several candidates.

The most straightforward solution of problem 1 would be evaluating $P(\mathbf{Y}|\mathbf{I}, \lambda) \times P(\mathbf{I}|\lambda)$ for a fixed states path $\mathbf{I}$=$\{i_1, i_2, \ldots, i_T\}$ and then summing up all the possible state paths $\mathbf{I}$.

$$\begin{aligned} P(\mathbf{Y}|\lambda) &= \sum_{all\ \mathbf{I}} P(\mathbf{Y}|\mathbf{I}, \lambda) \times P(\mathbf{I}|\lambda) \\ &= \sum_{i_1, i_2, \cdots, i_T} \pi_{i_1} b_{i_1}(Y_1) a_{i_1 i_2} b_{i_2}(Y_2) \cdots a_{i_{T-1} i_T} b_{i_T}(Y_T). \end{aligned} \tag{1.3}$$

It follows the complexity of this procedure is of order $2T \cdot N^T$. That means even for a small value of N and T, (e.g. N=5 T=200), the computation is still intractable for available computers. A more efficient algorithm then is called for.

Forward-Backward Method

Let define $\alpha_t(i)$ be the probability of partial observations up to time t and in state i at time t, given the HMM model λ:

$$\alpha_t(i) = P(\mathbf{Y}^{(t)}, X_t = i|\lambda), \tag{1.4}$$

where $\mathbf{Y}^{(t)}$ is the partial observation sequence up to time t, namely, $\mathbf{Y}^{(t)}$=$\{Y_1, Y_2, ... ,Y_t\}$.

Then

$$\begin{aligned} P(\mathbf{Y}|\lambda) &= P(\mathbf{Y}^{(T)}|\lambda) \\ &= \sum_{i=1}^{N} P(\mathbf{Y}^{(T)}, X_T = i|\lambda) \\ &= \sum_{i=1}^{N} \alpha_T(i). \end{aligned} \tag{1.5}$$

We can solve for $\alpha_T(i)$ inductively through the equation:

$$\begin{aligned} \alpha_t(j) &= P(\mathbf{Y}^{(t)}, X_t = j) \\ &= \sum_{i=1}^{N} P(Y_t, \mathbf{Y}^{(t-1)}, X_t = j, X_{t-1} = i) \\ &= \sum_{i=1}^{N} P(\mathbf{Y}^{(t-1)}, X_{t-1} = i) P(Y_t, X_t = j|\mathbf{Y}^{(t-1)}, X_{t-1} = i) \\ &= \sum_{i=1}^{N} P(\mathbf{Y}^{(t-1)}, X_{t-1} = i) P(X_t = j|\mathbf{Y}^{(t-1)}, X_{t-1} = i) P(Y_t|X_t = j, \mathbf{Y}^{(t-1)}, X_{t-1} = i) \\ &= \sum_{i=1}^{N} P(\mathbf{Y}^{(t-1)}, X_{t-1} = i) P(X_t = j|X_{t-1} = i) P(Y_t|X_t = j) \\ &= \sum_{i=1}^{N} [\alpha_{t-1}(i) \cdot a_{ij}] \cdot b_j(Y_t) \end{aligned} \tag{1.6}$$

and

$$\alpha_1(j) = P(Y_1, X_t = j) = \pi_j b_j(Y_1). \tag{1.7}$$

Often $\alpha_t(i)$ is referred as the *Forward Variable* and this method is called the *Forward Method*. Through this method, we achieve a computation complexity of order N^2T, a huge saving compared to $2T \cdot N^T$ of direct method.

Alternative to the forward method, there exists a *Backward Method* which is able to solve the problem. In a very similar manner, we define the *backward variable* $\beta_t(i) = P(\mathbf{Y}^{*(t)}|X_t = i, \lambda)$ where $\mathbf{Y}^{*(t)}$ denotes $\{Y_{t+1}, Y_{t+2}, \ldots, Y_T\}$, the partial time series beyond time t. Then we can use $\beta_t(i)$ to solve $P(\mathbf{Y}|\lambda)$ as easily as forward method:

Firstly we initialize $\beta_T(i)$,

$$\beta_T(i) = 1. \qquad\qquad 1 \leq i \leq N. \tag{1.8}$$

Then for $t = T-1, T-2, \cdots, 1$ and $1 \leq i \leq N$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) \cdot a_{ij} \cdot b_j(Y_{t+1}). \tag{1.9}$$

Finally,

$$P(\mathbf{Y}|\lambda) = \sum_{i=1}^{N} \pi_i b_i(Y_1) \beta_1(i). \tag{1.10}$$

The proof for (1.8)-(1.10) can be done in a very similar way to (1.5)-(1.7).

1.6 Solving Problem Two – Viterbi Algorithm

To solve problem 2, we have to find the optimal state sequences which could best explain the given observations in some way. The solutions for this problem rely on the optimality criteria we have chosen. The most widely used criterion is to maximize $P(\mathbf{Y}, \mathbf{X}|\lambda)$, which will be the case we discussed here. Again, the *observation and state probability* $P(\mathbf{Y}, \mathbf{X}|\lambda)$ is not a conditional probability. It represents the probability (for discrete distribution) or likelihood (for continuous distribution) of observing observation sequence $\mathbf{Y} = \{Y_1, Y_2, \cdots, Y_T\}$ and state sequence $\mathbf{X} = \{X_1, X_2, \cdots, X_T\}$ given their joint distribution $f(x, y)$.

Since the model $\lambda = (A, B, \pi)$ and the observation sequence $\mathbf{Y}$=$\{Y_1, Y_2, \ldots, Y_T\}$, the probability of the state path and observation sequence given the model would be:

$$\begin{aligned} P(\mathbf{Y}, \mathbf{X}|\lambda) &= P(\mathbf{Y}|\mathbf{X}, \lambda)P(\mathbf{X}|\lambda) \\ &= \pi_{X_1} b_{X_1}(Y_1) a_{X_1 X_2} b_{X_2}(Y_2) \cdots a_{X_{T-1} X_T} b_{X_T}(Y_T). \end{aligned} \tag{1.11}$$

To convert the products into summations, we define $U(\mathbf{X})$ as

$$\begin{aligned} U(\mathbf{X}) &= -\ln(P(\mathbf{Y}, \mathbf{X}|\lambda)) \\ &= -[\ln(\pi_{X_1} b_{X_1}(Y_1)) + \sum_{2}^{T} \ln(a_{X_{t-1} X_t} b_{X_t}(Y_t))]. \end{aligned} \tag{1.12}$$

Consequently,

$$\max_{\mathbf{X}} P(\mathbf{Y}, \mathbf{X}|\lambda) \Longleftrightarrow \min_{\mathbf{X}} U(\mathbf{X}).$$

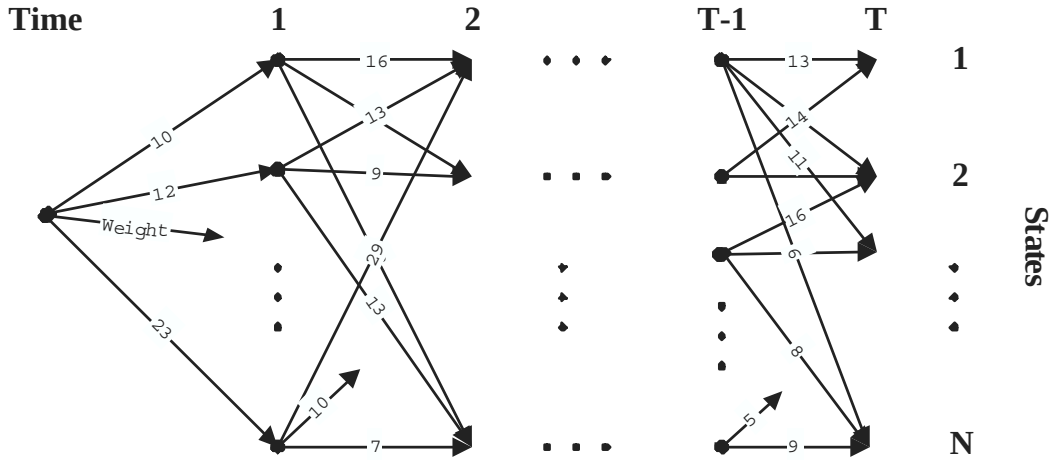

Figure 1.1 A graph of weighted pathes

This reformation now enables us to view terms like $-\ln(a_{X_{t-1}X_t}b_{X_t}(Y_t))$ as the cost (or distance) associated to the transition from state X_{t-1} to X_t. The problem then can be seen as finding the shortest path in a graph like (1.1). In the graph, the vertex corresponds to the states and the weight on the edge indicates the cost (or distance) between two vertexes.

Finding-the-shortest-path problem is one of the most fundamental problems in graph theory and can be solved by dynamic programming approaches, for example, *Viterbi Algorithm*.

Let $U_t(X_1, X_2, \cdots, X_t)$ be the first t terms of $U(\mathbf{X})$ and $V_t(i)$ be the minimal accumulated cost when we are in state i at time t,

$$U_t(X_1, X_2, \cdots, X_t) = -[\ln(\pi_{X_1}b_{X_1}(Y_1)) + \sum_{i=2}^{t} \ln(a_{X_{i-1}X_i}b_{X_i}(Y_i))], \quad (1.13)$$

$$V_t(i) = \min_{X_1, X_2, \cdots, X_{t-1}, X_t = i} U_t(X_1, X_2, \cdots, X_{t-1}, X_t = i). \tag{1.14}$$

Viterbi algorithm then can be implemented by four steps:

1. Initialize the $V_1(i)$ for all $1 \leq i \leq N$:

$$V_1(i) = -\ln(\pi_{X_i} b_{X_i}(Y_i)). \tag{1.15}$$

2. Inductively calculate the $V_t(i)$ for all $1 \leq i \leq N$, from time $t = 2$ to $t = T$:

$$V_t(i) = \min_{1 \leq j \leq N} [V_{t-1}(j)) - \ln(a_{X_j X_i} b_{X_i}(Y_i)]. \tag{1.16}$$

3. Then we get the minimal vale of $U(\mathbf{X})$:

$$\min_{\mathbf{X}} U(\mathbf{X}) = \min_{1 \leq i \leq N} [V_T(i)]. \tag{1.17}$$

4. Finally we trace back the calculation to find the optimal state path $\mathbf{X} = \{X_1, X_2, \cdots, X_T\}$.

1.7 Solving Problem Three – Baum-Welch Method

The third problem of HMM is to determine the parameters $\lambda = (A, B, \pi)$ based on the observation sequence $\mathbf{Y}$. Evaluating the parameters of HMM is not trivial. By far there is no analytical solution to this problem. The general approach is to train the model with the observation data using some iterative procedure until its convergence. More specifically, the parameter set $\lambda = (A, B, \pi)$ would be initialized with appropriate guesses at first. Then a set of re-estimation formula would be repeatedly

used in a number of iterations so that the parameter set could gradually approach to the ideal values where the occurrence possibility of the observation sequence are maximized.

Similar to the situation in problem 2, there are different criteria to interpret the problem. One criterion is the *maximum state optimized likelihood criterion* which tries to maximize $P(\mathbf{Y}, \mathbf{X}^*|\lambda)$ and the $\mathbf{X}^*$ here is the optimum state sequence as given by the solution in problem 2. Based on this criterion , we could use the *Segmental K-means Algorithm* to estimate the appropriate parameter set $\lambda = (A, B, \pi)$. We will discuss this algorithm in the next section. Another criterion is *maximum likelihood criterion* which tries to maximize $P(\mathbf{Y}|\lambda)$, the observation probability of Y given the parameter set. Based on this criterion, the problem could be solved by an iterative procedure *Baum-Welch Method*. We will focus on this method in the this section.

Baum-Welch Method

Baum-Welch method is indeed an implementation of general EM (Expectation-Maximization) method [5]. As indicated by its name, EM algorithm involves a two-step (E-step and M-step) procedure which will be recursively used. But before going into any details of EM algorithm, one need to define two variables in order to describe the algorithm mathematically.

Let $\xi_t(i, j)$ be the probability of the HMM being in state i at time t and making

a transition to state j at time $t+1$, given the model $\lambda = (A, B, \pi)$ and observation sequence $\mathbf{Y}$=$\{Y_1, Y_2, \ldots, Y_T\}$:

$$\xi_t(i,j) = P(X_t = i, X_{t+1} = j|\mathbf{Y}, \lambda). \tag{1.18}$$

Using Bayes law and the independency assumption we made before, it follows:

$$\begin{aligned}
\xi_t(i,j) &= \frac{P(X_t = i, X_{t+1} = j, Y|\lambda)}{P(Y|\lambda)} \\
&= \frac{P(X_t = i, \mathbf{Y}^{(t)}|\lambda)P(\mathbf{Y}^{*(t)}, X_{t+1} = j|X_t = i, \lambda)}{P(\mathbf{Y}|\lambda)} \\
&= \frac{P(X_t = i, \mathbf{Y}^{(t)}|\lambda)P(X_{t+1} = j|X_t = i)P(\mathbf{Y}^{*(t)}|X_{t+1} = j, X_t = i, \lambda)}{P(Y|\lambda)} \\
&= \frac{P(X_t = i, \mathbf{Y}^{(t)}|\lambda)P(X_{t+1} = j|X_t = i)P(Y_{t+1}|X_{t+1} = j, \lambda)P(\mathbf{Y}^{*(t+1)}|X_{t+1} = j, \lambda)}{P(Y|\lambda)} \\
&= \frac{\alpha_t(i)a_{ij}b_j(Y_{t+1})\beta_{t+1}(j)}{P(Y|\lambda)},
\end{aligned} \tag{1.19}$$

where forward variable $\alpha_t(i)$ and backward variable $\beta_t(i)$ follows the same definition in previous section:

$$\begin{aligned}
\alpha_t(i) &= P(\mathbf{Y}^{(t)}, X_t = i|\lambda) \qquad & \mathbf{Y}^{(t)} = \{Y_1, \cdots, Y_t\}, \\
\beta_t(i) &= P(\mathbf{Y}^{*(t)}|X_t = i, \lambda) \qquad & \mathbf{Y}^{*(t)} = \{Y_{t+1}, \cdots, Y_T\}.
\end{aligned}$$

We also define the $\gamma_t(i)$ as the probability in state i at time t given the observation sequence $\mathbf{Y}$ and model $\lambda = (A, B, \pi)$, then it can be proven:

$$\begin{aligned}
\gamma_t(i) &= P(X_t = i|\mathbf{Y}, \lambda) \\
&= \frac{P(X_t = i, \mathbf{Y}|\lambda)}{P(\mathbf{Y}|\lambda)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{P(X_t = i, \mathbf{Y}^{(t)}|\lambda) P(\mathbf{Y}^{*(t)}|X_t = i, \lambda)}{P(\mathbf{Y}|\lambda)} \\
&= \frac{\alpha_t(i)\beta_t(i)}{P(\mathbf{Y}|\lambda)}. \qquad (1.20)
\end{aligned}
$$

Note that

$$
\begin{aligned}
\sum_{t=1}^{T-1} \gamma_t(i) &= \textit{expected No. of transitions from state } i. \qquad (1.21) \\
\sum_{t=1}^{T-1} \xi_t(i,j) &= \textit{expected No. of transitions from state } i \textit{ to state } j. \qquad (1.22)
\end{aligned}
$$

With the above definition, then one can outline the Baum-Welch Re-estimation Formula:

$$
\begin{aligned}
\hat{\pi}_i &= \textit{expected frequency in state } i \textit{ at time } t = 1 \\
&= \gamma_1(i) \qquad (1.23) \\
\hat{a}_{ij} &= \frac{\textit{expected No. of transitions from state } i \textit{ to state } j}{\textit{expected No. of transitions from state } i} \\
&= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad (1.24) \\
\hat{b}_i(m) &= \frac{\textit{expected No. of times in state } i \textit{ and observating } V_m}{\textit{expected No. of times in state } i} \\
&= \frac{\sum_{t=1, Y_t = V_m}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}. \qquad (1.25)
\end{aligned}
$$

Equation (1.25) is in effect when the observations $\{Y_1, Y_2, \cdots, Y_T\}$ are discrete. In the case of continuous distribution, when $\{Y_1, Y_2, \cdots, Y_T\}$ are multivariate normal distributed, we are interested in the distribution parameters such as mean vector μ_i

and covariance matrix Σ_i when in state i,

$$\hat{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) Y_t}{\sum_{t=1}^{T} \gamma_t(i)} \tag{1.26}$$

$$\hat{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(Y_t - \hat{\mu}_i)(Y_t - \hat{\mu}_i)'}{\sum_{t=1}^{T} \gamma_t(i)}. \tag{1.27}$$

Suppose we have an initial guess of the parameters of HMM $\lambda_0 = (A_0, B_0, \pi_0)$ and several sequences of observations, we can use equation (1.21) and (1.22) to calculate the expected values of transition properties of the Markov Chain (the *Expectation* step of E-M algorithm). Then the maximum likelihood estimation of the model is computed through the recursive usage of equation (1.23)-(1.27) (the *Maximization* step of E-M algorithm).

Let $\hat{\lambda}_l$ be the parameter estimation in lth iteration. It can be proven[20] that either $\hat{\lambda}_l = \hat{\lambda}_{l-1}$ which means $\hat{\lambda}_l$ and $\hat{\lambda}_{l-1}$ reaches a critical point of the likelihood function, or $P(\mathbf{Y}|\hat{\lambda}_l) > P(\mathbf{Y}|\lambda_{l-1})$ which indicates that the observation sequences can be better explained by the new model $\hat{\lambda}_l$.

Based on the above procedure, the $\hat{\lambda}$ is iteratively re-estimated until it converges to a limit point. It should be remembered that Baum-Welch method leads to a local maximum of λ only.

In practice, to get a good solution, the initial guess λ_0 is very important. Usually several sets of starting guesses of λ_0 are used and one with the greatest likelihood value is chosen. Laird suggested a grid search method [20] which divides the searching

domain into small grids and starts from each of the intersections. Leroux and Puterman argues that the grid method would generate too many initial points when high dimensional space are involved. They suggests a clustering algorithm and a simply implementation can be found in [19].

1.8 Solving Problem Three – Segmental K-mean Algorithm

Segmental K-mean Algorithm (SKA) is another method widely used to estimate the parameter set $\lambda = (A, B, \pi)$ of hidden Markov models. Know from Baum-Welch method, SKA is based on the *maximum state optimized likelihood criterion*, in which one tries to maximize $L(\lambda|\mathbf{X}^*, \mathbf{Y})$, the likelihood function of λ given the optimal state sequence $\mathbf{X}^*$ and observation sequence $\mathbf{Y}$. Optimal state sequence $\mathbf{X}^*$ is actually the Viterbi path in most cases.

Like Baum-Welch method, the implementation of SKA also involves iterative procedures. In each iteration, it takes us from λ_l to λ_{l+1} such that $L(\lambda_{l+1}|\mathbf{X}^*_{l+1}, \mathbf{Y}) \geq L(\lambda_l|\mathbf{X}^*_l, \mathbf{Y})$ and eventually they will reach a local maximum.

Suppose there are N state symbols and a long observation sequence of length T . The main steps of the algorithm is as follows:

Step 1: Pick up N observations as the centroids of a cluster and assigns the rest of the $T - N$ observations to their nearest cluster based on their distance to those centroids. The distance is usually just the Euclidean distance.Those who falls into

the same cluster are assumed to belong to a same state and vice versa. The initial selection of centroids can be arbitrary but a good choice could greatly reduce the iterations needed for convergence. Another commonly used method is to divide the observation domain into N equally spaced segments and those falling into the same segments form an initial cluster.

Step 2: Estimate the initial probabilities $\hat{\pi} = [\hat{\pi}_i]$ and the transition probability $\hat{A} = [\hat{a}_{ij}]$:

$$\hat{\pi}_i = \frac{Number\ of\ occurrences\ of\ X_1 = i}{Number\ of\ observation\ sequence} \tag{1.28}$$

and

$$\hat{a}_{ij} = \frac{Number\ of\ transition\ from\ i\ to\ j}{Number\ of\ transition\ from\ i}. \tag{1.29}$$

Step 3: Calculate the distribution parameters related to B. For continuous multivariate Gaussian distribution, the mean vector and covariance matrix in state i, μ_i and Σ_i, can be estimated by:

$$\begin{aligned} \hat{\mu}_i &= \frac{\sum_{x_t=i} Y_t}{N_i} \\ \hat{\Sigma}_i &= \frac{1}{N_i} \sum_{x_t=i} (Y_t - \mu_i)'(Y_t - \mu_i), \end{aligned}$$

where N_i is the number of states i in the whole state sequence.

Step 4: Find the new optimal state sequence X^* based on new parameter set $\hat{\lambda} = (\hat{A}, \hat{B}, \hat{\pi})$ (Could use Vertibi path given in the solution of Problem 2).

Step 5: If there are any change in X^* , repeat step 2 to step 5.

It has already been proven[18] that SKA will converges to state optimized likelihood function for most commonly used distributions including the Gaussian distribution.

In a sense, E-M algorithm is somewhat better than SKA since it does not use $\mathbf{X}^*$, the estimated best state path as an input of the model. But in practice, though based on different criteria, the estimated parameters of those two are no much difference especially when a large number of parameters are to be estimated. Compared to E-M algorithm, SKA is usually easier to implement and more efficient when huge-amount data are involved because of the simpler form of its re-estimation formula.

1.9 H2M:Matlab Functions of HMM

H2M is a set of MATLAB functions which implement the EM algorithm to estimate the parameters of hidden Markov models. It is able to handle the multivariate HMM with a state-depended Gaussian distribution, as well as some discrete distributions such as Poisson distribution and negative binomial distribution.

A typical usage of H2M involves the following M-codes (MATLAB language) which well characterize the EM procedure in the case of state-depended Gaussian distribution:

```
for i = 1:n_iter
    [alpha, beta, logscale, dens] = hmm_fb(Y, A,pi0, mu, Sigma);
    logl(i) = log(sum(alpha(T,:))) + logscale;
    [A, pi0] = hmm_tran(alpha, beta, dens, A, pi0);
    [mu, Sigma] = hmm_dens(Y, alpha, beta, COV_TYPE);
end
```

In E-step, "hmm_fb" calculates the forward variables (alpha) and backward variables (beta) for the given observation sequence (Y) and initialization of parameters (A, pi0, mu, Sigma). Then the forward and backward variables are used to re-estimate the parameter set through functions "hmm_tran" and "hmm_dens" (M step). This E-M procedure are repeated until certain criteria are achieved (In above example, the E-M procedure are repeated for *n_iter* times which might not guarantee the convergence of the parameters.). Note in each iteration, as a by-product of forward variable, the log-likelihood values (logl(i)) of the current parameter set is stored which may be used as a good criteria for convergence.

As in the above example, the codes of H2M are quite straight-forward. Also in

the package there are a series of well-documented examples demonstrating its usage. The codes are readily implementable in the hidden Markov model set up using M-file programming.

In this section, we have provided a brief introduction to H2M. In the final chapter, an EM procedure will be implemented using H2M to compare the model adequacy with an autoregressive Markov model. Additional information can be found in the file h2m.pdf available with the H2M package.

Chapter 2
TIME SERIES ANALYSIS

2.1 Introduction of Stationary Time Series

A time series is a chronological sequence of observations on a variable of interest. The variable is observed at discrete time points, usually equally spaced. A mathematical description of the time sequence could be a sequence of random variables $\{x_t|\ t \in T\}$, where T is an index set of integers (say $\{1, 2, 3, \cdots\}$). The distribution of this sequence of random variables is specified by the joint distribution of every finite subsets of $\{x_t|\ t \in T\}$, say $\{x_{t_1}, x_{t_2}, \cdots, x_{t_k}\}$, for all integer k.

A time series $\{x_t|\ t \in T\}$ is *stationary* if the distribution of $\{x_{t_1}, x_{t_2}, \cdots, x_{t_k}\}$ is the same as the distribution of $\{x_{t_1+h}, x_{t_2+h}, \cdots, x_{t_k+h}\}$ for all choices of $\{t_1, t_2, \cdots, t_k\}$ and h such that $t_1, t_2, \cdots, t_k \in T$ and $t_1 + h, t_2 + h, \cdots, t_k + h \in T$. A time series which is not stationary is called *non-stationary*.

Broadly speaking, a time series is said stationary if there are no systematic change in the mean (no trend) and variance(equal breadth). More specifically, if a time series is stationary, it can be showed that its mean value function of is a constant and the autocorrelation between any two time points of the series depends only on the gap between them.

$$E[x_t] = \mu \tag{2.1}$$

$$Corr(x_t, x_{t+h}) = \sigma(h). \tag{2.2}$$

A time series satisfies above two conditions is *weakly stationary*. Note that the stationarity guarantees the weakly stationarity, but the converse is not true.

One of the simplest examples of stationary time series is a white noise series. $\{u_t|\ t \in T\}$ is a collection of identical-distributed and mutually independent random variables with common mean zero and constant variance σ^2. The stationarity of it is apparent. Actually, white noise timer series is a purely random process. It is called "white noise" because of the fact that it is very often been included in the more complicated probabilistic models(e.g. Moving-Average process) in engineering as the random error. Although we haven't specify its distribution here, in most cases it will be assumed to be normal distributed.

2.2 Some Time Series Models

2.2.1 Moving Average (MA) Processes

Suppose $\{u_t|\ t \in T\}$ is a white noise process with mean zero and variance σ^2. A process $\{x_t|\ t \in T\}$ said to be a moving average process of order q, written as MA(q), if

$$x_t = \mu + \alpha_0 u_t + \alpha_1 u_{t-1} + \alpha_2 u_{t-2} + \cdots + \alpha_q u_{t-q} \qquad t \in T \tag{2.3}$$

where $\{\alpha_i\}$ are constants. The u's are usually scaled so that $\alpha_0 = 1$.

It is easy to see that

$$E[x_t] \;=\; \mu + E[u_{t-1}] + \alpha_1 E[u_{t-1}] + \cdots + \alpha_q E[u_{t-q}] = \mu \tag{2.4}$$

and

$$\begin{aligned}
\sigma(t, t+h) &= Cov(x_t, x_{t+h}) \\
&= E((x_t - \mu)(x_{t+h} - \mu)) \\
&= \sum_{s=0}^{q} \sum_{l=0}^{q} \alpha_s \alpha_l E[u_{t-s} u_{t+h-l}] \\
&= \begin{cases} 0 & h > q \\ \sigma^2 \sum_{s=0}^{q-h} \alpha_s \alpha_{s+h} & h = 0, 1, \cdots, q \\ \sigma(t, t-h) & h < 0 \end{cases}
\end{aligned} \tag{2.5}$$

since

$$E(u_i u_j) = \begin{cases} \sigma^2 & i = j \\ 0 & i \neq j. \end{cases} \tag{2.6}$$

Because the autocorrelation function does not depend on the time t and the mean μ is a constant, the moving average process is weakly stationary. Furthermore, if the white noise $\{u_t |\ t \in T\}$ is normal distributed, it can be shown that the process is stationary.

As a special case of moving average process, MA(∞) satisfies the following equation:

$$x_t = \mu + \alpha_0 u_t + \alpha_1 u_{t-1} + \alpha_2 u_{t-2} + \cdots \tag{2.7}$$

where $\{u_t |\ t \in T\}$ is a white noise time series as usual. The mean function of the process is still the constant μ and the autocorrelation function is

$$\sigma(t, t+h) = \sigma^2 \sum_{s=0}^{\infty} \alpha_s \alpha_{s+h}. \tag{2.8}$$

It follows the MA(∞) is weakly stationary if $\sum_{s=0}^{\infty} |\alpha_s| < \infty$.

Let M be the linear space spanned by $\{x_t|\ t \in T\}$ (which can be called a Hilbert space). The *backshift operator* B is a mapping from M to itself , B:M$\rightarrow$M, and defined by B$x_t = x_{t-1}$. The backshift operator B provides another way to represent the MA(q) on the Hilbert space.

Note that $B^p x_t = x_{t-p}$. Then MA(q) and MA(∞) can be written respectively as:

$$x_t = \mu + \alpha(B)u_t \tag{2.9}$$

$$x_t = \mu + \theta(B)u_t \tag{2.10}$$

where $\alpha(B) = I + \alpha_1 B + \alpha_2 B^2 + \cdots + \alpha_q B^q$ and $\theta(B) = I + \theta_1 B + \theta_2 B^2 + \cdots$. These representations of MA(q) will facilitate our further discussion in the proceeding of the chapter.

2.2.2 Autoregressive (AR) Processes

Let $\{u_t|\ t \in T\}$ be a white noise process with mean zero and variance σ^2. A process $\{x_t|\ t \in T\}$ is said to be an autoregressive time series of order p, written as AR(p), if

$$x_t = \delta + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \cdots + \beta_p x_{t-p} + u_t \tag{2.11}$$

where $\{\beta_i\}$ are constants. The format of the AR process is rather like a multiple regression model. The prefix "auto" comes from the fact that x_t is regressed on the past values of itself. Another format of AR(p) is :

$$\beta(B)x_t = \delta + u_t \tag{2.12}$$

where $\beta(B) = I - \beta_1 B - \beta_2 B^2 - \cdots - \beta_p B^p$.

Let $\theta(B) = \beta^{-1}(B) = I + \theta_1 B + \theta_2 B^2 + \theta_3 B^3 + \cdots$, in which the relationship between βs and θs can be easily found. Then the equation (2.12) may be written as

$$\begin{aligned} x_t &= (\delta + u_t)/\beta(B) \\ &= (\delta + u_t)\theta(B) \\ &= \mu + u_t\theta(B) \\ &= \mu + u_t + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \theta_3 u_{t-3} + \cdots \end{aligned} \tag{2.13}$$

where the μ is a constant and can be calculated by

$$\mu = \delta/(1 - \beta_1 - \beta_2 - \cdots - \beta_p). \tag{2.14}$$

The equations show that x_t can be written as a infinite MA process, it follows that $E(x_t) = \mu$. And the autocovariance function is

$$\sigma(t, t+h) = \sigma^2 \sum_{s=0}^{\infty} \theta_s \theta_{s+h}. \tag{2.15}$$

A sufficient condition for its convergence and hence for stationarity, is that $\sum_{s=0}^{\infty} |\theta_s| < \infty$. An equivalent condition for stationarity is to say that the root of the polynomial $\beta(x) = 1 - \beta_1 x - \beta_2 x^2 - \cdots - \beta_p x^p$ must lie outside the unit circle[2].

Example: AR(1) process with $\delta = 0$

As a simple but important example, we look at the first-order case with $\delta = 0$. The process becomes:

$$x_t = \beta x_{t-1} + u_t.$$

When $|\beta| = 1$, x_t is called a random process and then

$$x_t = x_0 + \sum_{i=1}^{t} u_i.$$

It follows that $E(x_t) = 0$ and $Var(x_t) = Var(x_0) + t\sigma^2$. As the variance changes with t, the process is non-stationary.

When $|\beta| > 1$, since the $E(u_t) = 0$, the random term u_t will eventually disappear and thus the equation becomes:

$$x_t = \beta x_{t-1}.$$

Then the process will follow a non-stationary deterministic path.

Only when $|\beta| < 1$,

$$\begin{aligned} \mathrm{E}(x_t) &= 0 \\ \mathrm{Var}(x_t) &= \frac{\sigma^2}{1-\beta^2}. \end{aligned}$$

The process is stationary.

2.2.3 Mixed Autoregressive Moving Average(ARMA) Models

A useful class of time series is formed by combining MA and AR process. A mixed autoregressive moving average model containing p AR term and q MA term is a ARMA process of order (p,q) and it is given by:

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + \cdots + \beta_p x_{t-p} + \delta + u_t + \alpha_1 u_{t-1} + \alpha_2 u_{t-2} + \cdots + \alpha_q u_{t-q} \quad (2.16)$$

where $\{u_t |\ t \in T\}$ as usual, is a white noise time series. Apparently, the AR(p) and MA(q) processes we discussed in the previous two sections are degenerated cases of

ARMA(p,q) process. Using Back-Shift operator B, the formula can be simply written as:

$$\beta(B)x_t = \delta + \alpha(B)u_t \tag{2.17}$$

where

$$\beta(B) = I - \beta_1 B - \beta_2 B^2 - \cdots - \beta_p B^p \tag{2.18}$$

$$\alpha(B) = I + \alpha_1 B + \alpha_2 B^2 + \cdots + \alpha_q B^q. \tag{2.19}$$

Let

$$\psi(B) = \beta^{-1}(B)\alpha(B) = I + \psi_1 B + \psi_2 B^2 + \cdots \tag{2.20}$$

$$\phi(B) = \alpha^{-1}(B)\beta(B) = I - \phi_1 B - \phi_2 B^2 - \cdots \tag{2.21}$$

By multiplying equation (2.17) in both sides with $\beta^{-1}(B)$ and $\alpha^{-1}(B)$ respectively, we can get two different forms for ARMA(p,q) time series:

$$x_t = \mu + \phi(B)u_t \tag{2.22}$$

$$\psi(B)x_t = \nu + u_t \tag{2.23}$$

where μ and ν are two constants and can be calculated easily.

Equation (2.22) write the ARMA(p,q) process to the form of a pure MA(∞) process and sometimes referred as the *random shock form* of ARMA(p,q). Correspondingly, equation (2.23) is actually a pure AR(∞) and can be called the *inverted form* of it.

A little bit deeper understanding about the different forms of the ARMA(p,q) process would involve the dual relationship between AR(p) and MA(q) process. In short, a finite-order stationary AR(p) process corresponds to an infinite MA process and in turn, a finite stationary MA(q) process corresponds to an infinite MA process. This dual relationship also exists in the autocorrelation and partial autocorrelation functions.

2.2.4 Autoregressive Integrated Moving Average Models(ARIMA) and Box-Jenkins method

Most stationary time series can be modelled as a ARMA process, but in practice many time series, particularly those arising from economics and business area, are non-stationary. In order to apply the appropriate models discussed in the previous sections, non-stationary time series are often transformed into stationary ones. One widely used approach is to difference the series, i.e. replace the x_t in the equation (2.17): with $\nabla^d x_t$ where $\nabla = I - B$ and ∇^d denotes the dth difference. Then:

$$\beta(B)\nabla^d x_t = \delta + \alpha(B)u_t. \tag{2.24}$$

Such a model is called an *autoregressive integrated moving average* model of order (p,d,q) and abbreviated as *ARIMA(p,d,q)*.

For example, in the simple case ARIMA(0,1,1), the model actually is:

$$x_t = \delta + x_{t-1} + u_t + \alpha_1 u_{t-1}.$$

Since the autoregressive order is zero, it is also called *integrated moving average* of

order (1,1),or $IMA(1,1)$.

ARIMA process is capable of describing a class of non-stationary time series with a trend. It is developed as an central part of Box-Jenkins methodology. Box-Jenkins methodology provides a systematic procedure to identify an appropriate model for complex time series with trends, cycles, seasonal variations and even irregular fluctuations. The main approach is to examine the behaviors of sample autocorrelation function(SAC) and sample partial autocorrelation function(SPAC) of the time series under study. More can be found in Bowerman and O'Connell's [1].

2.3 Maximum Likelihood Estimation for ARMA models

The ARIMA model in the last section is essentially an natural extension of ARMA models. So in this section, we will describe the general method of finding the parameters of an ARMA(p,q) model.

The estimation approach is based on the *Maximum Likelihood Estimation* (MLE). Loosely speaking, the *likelihood* of a set data is the probability of obtaining that particular set of data, given the chosen probability distribution model. The *likelihood function* or its 'log' form (which is called the *log-likelihood function*) contains the unknown model parameters. The values of these parameters that maximize the sample likelihood are known as the *Maximum Likelihood Estimators*.

Follow the notations of last section, suppose the ARMA(p,q) has the form:

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + ... + \beta_p x_{t-p} + \delta + u_t + \alpha_1 u_{t-1} + \alpha_2 u_{t-2} + ... + \alpha_q u_{t-q}. \quad (2.25)$$

There are totally $p + q + 2$ parameters to be estimated.

To use the MLE, one needs to know the likelihood function $L(\ \beta_1, ..., \beta_p, \alpha_1, ..., \alpha_q, \delta, \sigma^2 | x_{1,} x_2, ..., x_T)$, or $L(\beta, \alpha, \delta, \sigma^2 | \mathbf{x})$ for short. Mathematically, the likelihood function is equal to the joint density function of $\mathbf{x}$, given the parameter set, $f(\mathbf{x}; \beta, \alpha, \delta, \sigma^2)$. This joint density function of $\mathbf{x}$ is not readily available because of the autoregressive structure of $\mathbf{x}$. However, if each white noise $\{u_1, u_2, ..., u_T\}$ is known as a function of parameter set $(\beta, \alpha, \delta, \sigma^2)$, the likelihood function can be calculated through equation (2.26) based on the fact that white noises $\{u_{1,} u_{2,}..., u_T\}$ are normally identical independent distributed (i.i.d.) with mean μ and variance σ^2:

$$\begin{aligned} L(\beta, \alpha, \delta, \sigma^2 | \mathbf{x}) &= f(x_1, x_2, \cdots, x_T; \beta, \alpha, \delta, \sigma^2) \\ &= f(u_{1,} u_{2,}..., u_T; \beta, \alpha, \delta, \sigma^2) \\ &= 2\pi^{-\frac{T}{2}} \sigma^{-T} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{t=1}^{T} u_t^2(\beta, \alpha, \delta, \sigma^2) \right\}. \end{aligned} \quad (2.26)$$

Since we know $\{x_t | t \in T\}$, if given the first q values for u_t, the whole white noise process $\{u_1, u_2, ..., u_T\}$ can be solved as a function of $\{\beta, \alpha, \delta, \sigma^2\}$ iteratively through equation (2.25). So the log-likelihood function is

$$l_x(\beta, \alpha, \delta, \sigma^2) = -\left(\frac{T}{2} \ln(2\pi) + \frac{T}{2} \ln(\sigma^2) + \frac{1}{2\sigma^2} \sum_{t=1}^{T} u_t^2(\beta, \alpha, \delta, \sigma^2) \right). \quad (2.27)$$

The maximization of $l_x(\beta, \alpha, \delta, \sigma^2)$ usually involved an iterative numerical procedure which will not be discussed here. Actually , nowadays most computer statistical packages could produce sound estimation with sophisticated routines.

2.4 Forecasting

Given all the parameters and the first T observation of an ARMA model, it is not difficult to make the forecasts. The lth step forecast $\hat{x}_T(l) = x_{T+l}$ is essentially a conditional expectation $E(x_{T+l}|x_T, x_{T-1}, ..., x_{t-p})$. To compute the forecasts, one should use the obvious fact:

$$\hat{x}_T(l) = x_T(l) \qquad if \quad l \leq 0 \tag{2.28}$$

and

$$\hat{u}_t(l) = \begin{cases} 0 & if \quad l > 0 \\ u_{t+l} & if \quad l \leq 0. \end{cases} \tag{2.29}$$

Recall in section 2.2.3 there are three forms of a ARMA model. Corresponding there are three forms of forecasting equation.

1. *Random shock form of the forecast*

   For the random shock form of ARMA Model (equation (2.22)), using equation(2.28) and (2.29), one would have:

$$\hat{x}_T(l) = \mu + \hat{u}_T(l) - \phi_1 \hat{u}_T(l-1) - \phi_2 \hat{u}_T(l-2) - ... \tag{2.30}$$

To obtain the forecast above, one need to compute all the error terms $\{u_T, u_{T-1}, ...\}$ from the observations $\{x_T, x_{T-1}, ...\}$ by iteratively using the equation:

$$u_t = x_t - \hat{x}_{t-1}(1). \tag{2.31}$$

Note $\hat{x}_0(1) = \mu$.

From equation (2.30), we could directly get the errors of forecasts:

$$\begin{aligned} e_T(l) &= x_{t+l} - \hat{x}_T(l) \\ &= u_{T+l} - \phi_1 u_{T+l-1} - \phi_2 u_{T+l-2} - \phi_{l-1} u_{T+1}. \end{aligned} \tag{2.32}$$

So the mean square error (MSE) for the l step forecasts can be calculated as:

$$\begin{aligned} MSE &= E[(u_{T+l} - \phi_1 u_{T+l-1} - \phi_2 u_{T+l-2} - \phi_2 u_{T+1})^2] \\ &= (1 + \phi_1^2 + \phi_2^2 + ... + \phi_{l-1}^2)\sigma^2. \end{aligned} \tag{2.33}$$

Hence

$$\sigma_T(l) = \sigma\sqrt{1 + \phi_1^2 + \phi_2^2 + ... + \phi_{l-1}^2}. \tag{2.34}$$

So the $(1-\alpha)100\%$ confidence interval for prediction x_{T+l} are given by

$$\left(x_T(l) - Z_{\alpha/2}\sigma_T(l), x_T(l) - Z_{\alpha/2}\sigma_T(l)\right).$$

2. *Inverted form of the forecast*

   Using equation (2.23), the invert form of the forecast is:

$$\hat{x}_T(l) = \upsilon + -\psi_1 \hat{x}_T(l-1) - \psi_2 \hat{x}_T(l-2) - \cdots \tag{2.35}$$

3. *Difference equation form of the forecast*

$$\hat{x}_T(l) = (1+\beta_1)\hat{x}_T(l-1) - \beta_2 \hat{x}_T(l-2) + \hat{u}_T(l) + \alpha_1 \hat{u}_T(l-1) + \alpha_2 \hat{u}_T(l-2). \tag{2.36}$$

Although those three predictions would give exactly the same point predictions, the random shock form are most commonly used because its coefficients could be directly used in the computation of the confidence limits.

The above forecasting formula are based on the Boxs-Jenkins ARIMA models [2]. But it should be mentioned that there are many other forecasting methods available and research shows no one could claim itself as the “best” method.

Chapter 3
AUTOREGRESSIVE HIDDEN MARKOV MODELS

3.1 Introduction

A time series may sometimes consist of observations generated by different mechanisms at different times. When this happens, the time series observations would act like switching back and forth between a couple of distinct states. When changing into a different state, the time series may have a significant change in their means or in their frequencies or breadthes of their fluctuations. The *Autoregressive Hidden Markov model*(*ARHMM*) are often being used to deal with this kind of time series. As indicated by the name, an ARHMM is the combination of an autoregressive time series model and a hidden Markov model. The autoregressive structure admits the existence of dependency amongst time series observations while the hidden Markov chain could capture the probability characteristics of the transitions amongst the underlying states. Actually, ARHMM is also referred as *time series with change in regime*(or *states*) by the econometricians.

To be more specific, let us see an example of ARHMM. As usual, $\mathbf{Y} = \{Y_1, Y_2, ..., Y_T\}$ denote the observation sequence. Each Y_t is a observation vector with k component $Y_t = \{y_1, y_2, ..., y_k\}'$. $\mathbf{X} = \{X_1, X_2, ..., X_T\}$ is a hidden state sequence with N possible states. $\mathbf{X}$ is assumed to be a Markov chain with transition matrix $A = [a_{ij}]$ and

initial distribution vector $\pi = [\pi_i]$.

As indicated earlier, the observation sequence $\mathbf{Y} = \{Y_1, Y_2, ..., Y_T\}$ is an AR(p) process which can be written as:

$$Y_t = \beta_0^{(X_t)} + \beta_1^{(X_t)} Y_{t-1} + \beta_2^{(X_t)} Y_{t-2} + ... + \beta_p^{(X_t)} Y_{t-p} + \varepsilon_t \tag{3.1}$$

or

$$Y_t = S_t \beta^{(X_t)} + \varepsilon_t \tag{3.2}$$

where

$$\begin{aligned} S_t &= (1, Y_{t-1}, Y_{t-2}, ..., Y_{t-p}) \\ \beta^{(X_t)} &= (\beta_0^{(X_t)}, \beta_1^{(X_t)}, \beta_2^{(X_t)}, ...\beta_p^{(X_t)})' \\ \varepsilon_t &\sim \quad i.i.d \;\; N(0, \Sigma). \end{aligned}$$

$\beta_i^{(X_t)}$ is the ith parameter for the autoregressive process when in state X_t. So the current observation Y_t are not only depends on the last p observations, but also the current states. In this example, the white noise ε_t are independent identical distributed with mean zero and covariance matrix Σ. But it should be mentioned that the ARHMM with heteroskedasticity (unequal variance) for distinct state X_t could also be developed with more complexity. In such cases, the error term ε_t will usually be replaced by ε_{X_t} which depended on the value of current state X_t. For the reason of computational tractability, we are not going into this issue in this thesis.

Another notation we have to make is the state-related probability distribution of the observations $B = [(b_j(Y))]$. In the previous chapter, we have used probability mass function for those discrete distribution. Now we will introduce the the probability density function (pdf) for the continuous case. The most general form of pdf in AR-HMM is of a finite mixture form:

$$b_j(Y) = \sum_{m=1}^{M} C_{jm} \Psi[Y, \mu_{jm}, \Sigma_{jm}] \tag{3.3}$$

where

Y is the observation vector being modelled.

C_{jm} is the m^{th} mixture coefficient in state j. Note C_{jm} 's are non-negative and satisfy the stochastic constraint:

$\sum_{m=1}^{M} C_{jm} = 1$ for all $1 \leq j \leq N$.

Ψ is any log-concave or elliptically symmetric density (e.g. Gaussian density).

μ_{jm}, Σ_{jm} are the mean and covariance vector for the m^{th} mixture density in state j, respectively.

As a special case of this class of mixture distribution , single component ($M = 1$) Gaussian density AR(p)-HMM would have the mean vector $S_t' \beta^{(X_t)}$ and covariance matrix $\sigma^2 I_{k*k}$, with a pdf:

$$P(Y|Z_t,\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{(Y_t - S_t'\beta^{(X_t)})'(Y_t - S_t'\beta^{(X_t)})}{2\sigma^2}] \tag{3.4}$$

where

θ is the parameter set with respect to the probability densities $B = \{b_{jm}\}$:

$\theta = \{\sigma^2, \delta', \beta_1', \beta_2', ...\beta_p')$. $\delta = (\delta^1, \delta^2, ..., \delta^N)$ and $\beta_i = (\beta_i^1, \beta_i^2, ...\beta_i^N)$.

Z_t is the information set for the latest $p+1$ states and latest past p observations,

$Z_t = \{X_{t-p}, ..., X_{t-2}, X_{t-1}, X_t, Y_{t-p}, ..., Y_{t-2}, Y_{t-1}\}$.

We will discuss this model in detail in the following sections.

With the structure described above , the parameters of the mixture AR(p)-HMM include:

1. the transition matrix matrix $A = [a_{ij}]$, $i, j = 1, 2, ..., N$;
2. the initial probability $\pi = [\pi_i]$, $i = 1, 2, ..., N$;
3. the matrix weight $C = [C_{jm}]$, $j = 1, 2, ..., N$; $m = 1, 2, ..., M$;
4. all necessary parameters defining the set of probability densities $\{b_{jm}\}$, $\theta = \{\sigma^2, \delta', \beta_1', \beta_2', ...\beta_p')$.

3.2 Juang and Rabiner's Estimation of ARHMM

In 1980's, B.H.Juang and L.R.Rabiner published a series papers regarding the application of HMM to the speech recognition. A class of HMMs particularly applicable

to speech processing, namely the finite Gaussian mixture autoregressive HMMs have been discussed in their papers. The corresponding estimation algorithms are also developed and applied to their speech recognizers. In this section , we will introduce and discuss their estimation algorithms of ARHMM.

For convenience , we use another version of equations (3.1) for AR(p) process:

$$Y_t = -\sum_{i=1}^{p} \beta_i Y_{t-i} + \varepsilon_t \tag{3.5}$$

where $\varepsilon_t \sim$ i.i.d N$(0, I)$

Note the unity variance assumption of ε_t implies the observation sequence $Y = \{Y_1, Y_2, ..., Y_T\}$ have already been normalized. This has been done by dividing each sample by $\sqrt{T\sigma^2}$, where T denotes the length of the observation sequence and σ^2 is the sample variance of the observations.

It can be shown [16][17] that for large T, the density function for $\mathbf{Y}$ is approximately

$$f(\mathbf{Y}) \simeq (\mathbf{2}\pi)^{-\frac{\mathbf{T}}{\mathbf{2}}} \exp\{-\frac{\mathbf{1}}{\mathbf{2}}\delta(\mathbf{Y};\beta)\} \tag{3.6}$$

where

$$\begin{aligned} \delta(Y;\beta) &= r_\beta(0)r(0) + 2\sum_{i=1}^{p} r_\beta(i)r(i) \\ \beta &= [1, \beta_1, \beta_2, ..., \beta_p]' \end{aligned}$$

$$
\begin{aligned}
r_\beta(i) &= \sum_{n=0}^{p-i} \beta_n \beta_{n+i} \quad with \quad \beta_0 = 1 \\
r(i) &= \sum_{n=0}^{t-i-1} Y_n Y_{n+i}.
\end{aligned}
$$

Note r_β's are the autocorrelations of the autoregressive coefficient and r's are the autocorrelation of the normalized observation samples. With this approximation, the density is defined by an autoregressive vector β or equivalently an autocorrelation vector $r_\beta = [r_\beta(0), r_\beta(1), ..., r_\beta(p)]$.

As a specific realization of equation (3.3) , we also assume the ARHMM is of a finite mixture form

$$b_j(Y) = \sum_{m=1}^{M} C_{jm} b_{jm}(Y) \tag{3.7}$$

for which $b_{jm}(Y)$ is a Gaussian p.d.f. Then it follows equation (3.7) can be approximated as:

$$b_{jm}(Y) \simeq (2\pi)^{-\frac{T}{2}} \exp\{-\frac{1}{2}\delta(Y; \beta_{jm})\} \tag{3.8}$$

where β_{jm} is the parameter vector defining the density for the mth mixture component in state j.

The estimation procedure of Juang and Rabiner are also based on E-M algorithm. It begins with an initial guess of model $\lambda = (A, \pi, C, \theta)$. Based upon λ , a training procedure is implemented which would lead to new model $\widehat{\lambda}$. The new model $\widehat{\lambda}$ will be better than the old one in the sense that $P(Y|\widehat{\lambda}) \geq P(Y|\lambda)$. After replacing the

old model λ with the new model $\widehat{\lambda}$, the procedure is iterated until a critical point is achieved.

Here I will just outline the re-estimation formula for the model parameter set. The deduction and the proof for convergence could refer to [19][20][27]:

1. The transition matrix $A = [a_{ij}]$, $1 \leq i, j \leq N$:

$$\widehat{a}_{ij} = \frac{\sum_{t=1}^{T} f(Y, X_{t-1} = i, X_t = j|\lambda)/f(Y|\lambda)}{\sum_{t=1}^{T} f(Y, X_{t-1} = i|\lambda)/f(Y|\lambda)}. \tag{3.9}$$

2. The mixture weight $C = [c_{jm}]$, $1 \leq j \leq N$, $1 \leq m \leq M$:

$$\widehat{c}_{jm} = \frac{\sum_{t=1}^{T} f(Y, X_t = j, h_t = m|\lambda)/f(Y|\lambda)}{\sum_{t=1}^{T} f(Y, X_t = j|\lambda)/f(Y|\lambda)} \tag{3.10}$$

where $h_t \in \{1, 2, ..., M\}$ is a random variable and denote the event that Y_t is drawn from the mixture component h_t.

3. Let r_{jm} represent the autocorrelation parameters for each mixture m in state j, $1 \leq m \leq M$, $1 \leq j \leq N$. r_{jm}'s can be used to calculate the β_{jm} in equation (3.8) and their re-estimation formulas are:

$$\widehat{r}_{jm}(i) = \frac{\sum_{t=1}^{T} f(Y, X_t = j, h_t = m|\lambda) \cdot r_t(i)/f(Y|\lambda)}{\sum_{t=1}^{T} f(Y, X_t = j, h_t = m|\lambda)/f(Y|\lambda)} \tag{3.11}$$

for $i = 0, 1, 2, ..., p$, $j = 1, 2, ..., N$ and $m = 1, 2, ...M$.

where $r_t(i) = \sum_{j=0}^{k-1+i} y_{t,j}y_{t,j+1}$ and $y_t = [y_{t,0}, y_{t,1}, ..., y_{t,k-1}]'$.

To effectively calculate the likelihood function $f(\cdot)$, we still adopt the backward and forward variables $\beta_t(\cdot)$ and $\alpha_t(\cdot)$ defined in chapter 1 :

$$\begin{aligned} \alpha_t(j) &= P(Y^{(t)}, X_t = j|\lambda) \\ \beta_t(j) &= P(Y^{*(t)}|X_t = j, \lambda). \end{aligned}$$

Then it is not very difficult to see,

$$\begin{aligned} f(Y, X_t = j|\lambda) &= \alpha_t(j)\beta_t(j) \\ f(Y|\lambda) &= \sum_{j=1}^{N} \alpha_T(j) \\ f(Y, X_{t-1} = i, X_t = j|\lambda) &= \alpha_{t-1}(i)a_{ij}b_j(Y_t)\beta_t(j) \\ f(Y, X_{t-1} = i, h_t = m|\lambda) &= \sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij}c_{jm}b_{jm}(Y_t)\beta_t(j). \end{aligned}$$

3.3 E-M Algorithm

In this section, I will briefly describe the theory behind the E-M algorithm and its properties. E-M algorithm was originally designed to deal with the missing values in the time series analysis. The unknown states in the HMM can be seen as the missing values in the E-M algorithm.

Followed the usual notations, let $\mathbf{Y}$= $[Y_1, Y_2, ..., Y_t]$ be the observation sequence,

$\mathbf{X}$= $[X_1, X_2, ..., X_t]$ be the unknown state sequence and $\lambda = (\pi, A, B)$ be the parameter set.

The goal is to maximize the observation probability $P(\mathbf{Y}|\lambda)$ by choosing appropriate λ. Mathematically $P(\mathbf{Y}|\lambda)$ is equivalent to the likelihood function of Y with unknown parameter set λ and it can be written as

$$P(\mathbf{Y}|\lambda) = \int_{\mathbf{X}} P(\mathbf{Y}, \mathbf{X}|\lambda) = \sum_{X_t=1}^{N} \sum_{X_{t-1}=1}^{N} ... \sum_{X_1=1}^{N} P(\mathbf{Y}, X_1, ..., X_t|\lambda). \tag{3.12}$$

In this way, the observation likelihood is parameterized in terms of $P(\mathbf{Y}, \mathbf{X}|\lambda)$. It will prove useful to define a new expression $Q(\lambda_{l+1}; \lambda_l, \mathbf{Y})$, the expected log-likelihood, where the log-likelihood is parameterized by λ_{l+1} and the expectation is taken with respect to another parameter set λ_l :

$$Q(\lambda_{l+1}; \lambda_l, \mathbf{Y}) = \int_{\mathbf{X}} \log(P(\mathbf{Y}, \mathbf{X}|\lambda_{l+1}))P(\mathbf{Y}, \mathbf{X}|\lambda_l). \tag{3.13}$$

The E-M algorithm starts from an initial guess of parameter set $\widehat{\lambda}_0$, then we can iteratively solve $\widehat{\lambda}_{l+1}$ $(l = 0, 1, ...)$ for the equation that maximizes $Q(\lambda_{l+1}; \lambda_l, Y)$:

$$\int_{\mathbf{X}} \frac{\partial \log P(\mathbf{Y}, \mathbf{X}|\lambda_{l+1})}{\partial \lambda_{l+1}}|_{\lambda_{l+1}=\widehat{\lambda}_{l+1}} \cdot P(\mathbf{Y}, \mathbf{X}|\widehat{\lambda}_l) = 0. \tag{3.14}$$

Then it is not very difficult to prove ([9][21]) the following two properties of E-M algorithm:

Proposition 1:

$$P(\mathbf{Y}|\widehat{\lambda}_{l+1}) \geq P(\mathbf{Y}|\widehat{\lambda}_l)$$

with equality only when $\widehat{\lambda}_{l+1} = \widehat{\lambda}_l$.

Proposition 2:

If

$$\frac{\partial Q(\lambda_{l+1}; \widehat{\lambda}_l, \mathbf{Y})}{\partial \lambda_{l+1}}\Big|_{\lambda_{l+1}=\widehat{\lambda}_l} = 0$$

then

$$\frac{\partial P(\mathbf{Y}|\lambda)}{\partial \lambda}\Big|_{\lambda=\widehat{\lambda}_l} = 0.$$

The first proposition claims that each iteration of E-M algorithm ensures an increased (or equal) value of likelihood function. The second proposition demonstrates that the sequence $\left\{\widehat{\lambda}_l\right\}_{l=1}^{\infty}$ converges to the local MLE. These two propositions together justify why the E-M algorithm yields the maximum likelihood estimate $\widehat{\lambda}$.

With $\lambda = (A, B, \pi)$, J.Hamilton [10] showed how equation (3.14) can be solved for A, B and π and hence we get a particular form of the E-M algorithm for the AR-HMM:

$$a_{ij}^{(l+1)} = \frac{\sum_{t=p+1}^{T} P(X_t = j, X_{t-1} = j|Y; \lambda_l)}{\sum_{t=p+1}^{T} P(X_{t-1} = j|Y; \lambda_l)} \tag{3.15}$$

$$\sum_{t=p+1}^{T} \sum_{X_t=1}^{N} \sum_{X_{t-1}=1}^{N} ... \sum_{X_{t-p}=1}^{N} \frac{\partial \log P(Y_t|Z_t; \theta)}{\partial \theta}\Big|_{\theta=\theta_{l+1}} P(X_t, ..., X_{t-p}|Y; \lambda_l) = 0 \tag{3.16}$$

$$\pi_{i_p, i_{p-1}, ..., i_1}^{(l+1)} = P(X_p = i_p, X_{p-1} = i_{p-1}, ..., X_1 = l_1|Y; \lambda_l) \qquad i_1, ..., i_p = 1, 2, ...N \tag{3.17}$$

where $Z_t = \{X_t, X_{t-1}, ..., X_{t-p}, Y_{t-1}, Y_{t-2}, ..., Y_{t-p}\}'$.

In each iteration , we calculate the smoothed probabilities such as $P(X_t, ..., X_{t-p}$ $|Y; \lambda_l)$ and then solves the $\lambda_{l+1} = (A^{(l+1)}, B^{(l+1)}, \pi^{(l+1)})$ as a function of previous estimation λ_l . The calculation of equation (3.15) and equation (3.17) is quite straightforward. Actually we'll see the differential in equation (3.16) often has a simpler form.

For example the Baum-Welch re-estimation formula (equations 1.23-1.25) in chapter 1 is essentially a special case of equations (3.15)-(3.17) with the autoregressive order $p = 0$.

Consider the case when the Y_t is i.i.d. Gaussian distributed with the mean vector and covariance matrix depending on the current state X_t :

$$Y_t \sim N(\mu_{X_t}, \Sigma_{X_t}).$$

The p.d.f. can be written as

$$P(Y_t|Z_t; \lambda) = \frac{1}{(2\pi)^{n/2}|\Sigma_{X_t}|^{1/2}} \exp[\frac{-(Y_t - \mu_{X_t})'\Sigma_{X_t}^{-1}(Y_t - \mu_{X_t})}{2}]. \tag{3.18}$$

So the differential part of equation (3.16) would be:

$$\begin{aligned} \frac{\partial \log P(Y_t|Z_t;\theta)}{\partial \mu_j} &= \Sigma_j^{-1}(Y_t - \mu_{X_t}) \quad if \quad X_t = j \\ &= 0 \qquad\qquad\qquad otherwise \end{aligned}$$

$$\begin{aligned} \frac{\partial \log P(Y_t|Z_t;\theta)}{\partial \Sigma_j^{-1}} &= \frac{1}{2}\Sigma_j - \frac{1}{2}(Y_t - \mu_j)(Y_t - \mu_j)' \quad if \quad X_t = j \\ &= 0 \qquad\qquad\qquad\qquad\qquad otherwise. \end{aligned}$$

Thus the equation (3.16) would have the form ($p = 0$):

$$\sum_{t=1}^{T}[\Sigma_j^{(l+1)}]^{-1}(Y_t - \mu_j^{(l+1)}) \cdot P(X_t = j|Y; \lambda_l) = 0 \tag{3.19}$$

$$\sum_{t=1}^{T}\Big[\frac{1}{2}\Sigma_j^{(l+1)} - \frac{1}{2}(Y_t - \mu_j^{(l+1)})(Y_t - \mu_j^{(l+1)})'\Big] \cdot P(X_t = j|Y; \lambda_l) = 0 \tag{3.20}$$

for $j = 1, 2, ...N$.

Solve for $\Sigma_j^{(l+1)}$ and $\mu_j^{(l+1)}$, we have

$$\mu_j^{(l+1)} = \frac{\sum_{t=1}^{T} Y_t \cdot P(X_t = j|Y; \lambda_l)}{\sum_{t=1}^{T} P(X_t = j|Y; \lambda_l)} \qquad j = 1, 2, ..., N \tag{3.21}$$

$$\Sigma_j^{(l+1)} = \frac{\sum_{t=1}^{T}(Y_t - \mu_j^{(l+1)})(Y_t - \mu_j^{(l+1)})' P(X_t = j|Y; \lambda_l)}{\sum_{t=1}^{T} P(X_t = j|Y; \lambda_l)} \tag{3.22}$$

which explains where the equation (1.26) and (1.27) come from.

3.4 E-M Formula for ARHMM

Now it comes to the estimation procedure of the ARHMM. Basically we will follow the structure and notations in the Section (3.1). Recall the autoregressive structure of the observation vectors have been parameterized as:

$$Y_t = S_t'\beta^{(X_t)} + \varepsilon_t \tag{3.23}$$

where

$$\begin{aligned} S_t &= (1, Y_{t-1}, Y_{t-2}, ..., Y_{t-p}) \\ \beta^{(X_t)} &= (\delta^{(X_t)}, \beta_1^{(X_t)}, \beta_2^{(X_t)}, ...\beta_p^{(X_t)}) \\ \varepsilon_t &\sim i.i.d \;\; N(0, \sigma^2). \end{aligned}$$

Then the conditional p.d.f of Y_t can be written as:

$$P(Y_t|Z_t;\lambda) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-(Y_t - S_t'\beta^{(X_t)})^2}{2\sigma^2}\right]. \tag{3.24}$$

To get the specific estimation formula, differentiate 3.24 with respect to β_j and σ^{-2}:

$$\begin{aligned} \frac{\partial \log P(Y_t|Z_t;\theta)}{\partial \beta_j} &= \frac{(Y_t - S_t'\beta_j)S_t}{\sigma^2} \qquad if \;\; X_t = j \\ &= 0 \quad otherwise \\ \frac{\partial \log P(Y_t|Z_t;\theta)}{\partial \sigma^{-2}} &= \frac{\sigma^2}{2} - \frac{(Y_t - S_t'\beta_j)^2}{2} \quad if \;\; X_t = j \\ &= 0 \quad otherwise. \end{aligned}$$

Then the equation (3.16) can be written as:

$$\sum_{t=p+1}^{T} \frac{(Y_t - S_t'\beta_j^{(l+1)})S_t}{\sigma_{(l+1)}^2} P(X_t = j|Y;\lambda_l) = 0 \tag{3.25}$$

$$\sum_{t=p+1}^{T}\sum_{j=1}^{N}\left[\frac{\sigma_{(l+1)}^2}{2} - \frac{(Y_t - S_t'\beta_j^{(l+1)})^2}{2}\right] P(X_t = j|Y;\lambda_l) = 0. \tag{3.26}$$

The estimation of $\beta_j^{(l+1)}$ which solves equation (3.25) can be found from an *ordinary least square* (OLS) regression of $\widetilde{Y}_t(j)$ and $\widetilde{S}_t(j)$:

$$\beta_j^{(l+1)} = \left[\sum_{t=p+1}^{T} \widetilde{[S}_t(j)][\widetilde{S}_t(j)]'\right]^{-1} \left[\sum_{t=p+1}^{T} \widetilde{[S}_t(j)]\widetilde{Y}_t(j)\right] \tag{3.27}$$

where

$$\widetilde{Y}_t(j) = Y_t \cdot \sqrt{P(X_t = j|Y; \lambda_l)}$$

$$\widetilde{S}_t(j) = S_t \cdot \sqrt{P(X_t = j|Y; \lambda_l)}$$

and thus the estimation of $\sigma^2_{(l+1)}$ is:

$$\sigma^2_{(l+1)} = \sum_{t=p+1}^{T} \sum_{j=1}^{N} \frac{(\widetilde{Y}_t(j) - \widetilde{S}_t(j)\beta_j^{(l+1)})^2}{T-p}. \tag{3.28}$$

The estimation of the transition probabilities $A = [a_{ij}]$ and the initial probabilities $\pi = [\pi_j]$ come from the eqn(3.15) and eqn(3.17):

$$a_{ij}^{(l+1)} = \frac{\sum_{t=p+1}^{T} P(X_t = j, X_{t-1} = j|Y; \lambda_l)}{\sum_{t=p+1}^{T} P(X_{t-1} = j|Y; \lambda_l)}$$

and

$$\pi_j^{(l+1)} = P(X_p = j|Y; \lambda_l), j = 1, 2, ..., N-1$$

and $\pi_N^{(l+1)} = 1 - \pi_1^{(l+1)} - \pi_2^{(l+1)} - ... - \pi_{N-1}^{(l+1)}$.

3.5 The Calculation of the Smoothed Probabilities

Every iteration of the re-estimation formula in the last section involves the calculation of the smoothed probabilities such as $P(X_t, X_{t-1}|Y)$ and $P(X_t|Y)$. Recall in chapter 1 how we use forward variable and backward variable to effectively calculate those probabilities for conventional hidden Markov models. When it comes to the ARHMM case, the principles are essentially the same. But the implementation of the calculation are inevitably more complex due to the autoregressive structure. In this section we will outline the iterative procedures of calculation of general smoothed probability $P(X_t, X_{t-1}, ..., X_{t-p}|Y)$, where p is the autoregressive order as usual.

1. The start-up of the algorithm needs to initialize the following two probabilities:

$$P(Y_{p+1}|Y^{(p)}) = \sum_{X_{p+1}=1}^{N} \sum_{X_p=1}^{N} ... \sum_{X_1=1}^{N} P(X_{p+1}|X_p) \cdot P(Y_{p+1}|Z_{p+1}) \pi_{X_p,...,X_1} \tag{3.29}$$

$$P(X_{p+1}, ..., X_1|Y^{(p+1)}) = \frac{P(X_{p+1}|X_p) \cdot P(Y_{p+1}|Z_{p+1}) \pi_{X_p,...,X_1}}{P(Y_{p+1}|Y^{(p)})} \tag{3.30}$$

Where

$$\begin{aligned} Y^{(t)} &= (Y_1, Y_2, ..., Y_t)' \\ Z_t &= \{X_t, X_{t-1}, ..., X_{t-p}, Y_{t-1}, Y_{t-2}, ..., Y_{t-p}\}' \\ \pi_{X_p,X_{p-1},...,X_1} &= P(X_p, X_{p-1}, ..., X_1|Y). \end{aligned}$$

2. Compute all the $P(Y_t|Y^{(t-1)})$ and $P(X_t, X_{t-1}, ..., X_{t-p}|Y^{(t)})$ for $t = p+2, ..., T$ by the formula:

$$P(Y_t|Y^{(t-1)}) = \sum_{X_t=1}^{N} \sum_{X_{t-1}=1}^{N} ... \sum_{X_{t-p-1}=1}^{N} P(X_t|X_{t-1}) \cdot P(Y_t|Z_t) \cdot P(X_{t-1}, ..., X_{t-p-1}|Y^{(t-1)}) \tag{3.31}$$

$$P(X_t, ..., X_{t-p}|Y^{(t)}) = \frac{\sum_{X_{t-p-1}=1}^{N} P(X_t|X_{t-1}) \cdot P(Y_t|Z_t) \cdot P(X_{t-1}, ..., X_{t-p-1}|Y^{(t-1)})}{P(Y_t|Y^{(t-1)})}. \tag{3.32}$$

3. For a particular fixed t , evaluate the advanced probability for $\tau = t+1, t+2, ..., t+p$:

$$P(X_\tau, ..., X_{t-p}|Y^{(\tau)}) = \frac{P(X_\tau|X_{\tau-1}) \cdot P(Y_\tau|Z_\tau) \cdot P(X_{\tau-1}, ..., X_{t-p}|Y^{(\tau-1)})}{P(Y_\tau|Y^{(\tau-1)})}. \tag{3.33}$$

4. Carry forward the inference for $\tau = t+p+1, t+p+2, ..., T$:

$$P(X_\tau, ..., X_{\tau-p}, X_t, ..., X_{t-p}|Y^{(\tau)}) = \tag{3.34}$$
$$\frac{\sum_{X_{\tau-p-1}=1}^{N} P(X_\tau|X_{\tau-1}) \cdot P(Y_\tau|Z_\tau) \cdot P(X_{\tau-1}, ...X_{\tau-p-1}, X_t, ..., X_{t-p}|Y^{(\tau-1)})}{P(Y_\tau|Y^{(\tau-1)})}.$$

5. Finally, we could finish the calculation of the smoothed probabilities by summing up the last p states :

$$P(X_t, X_{t-1,}..., X_{t-p}|Y) = \sum_{X_T=1}^{N} \sum_{X_{T-1}=1}^{N} ... \sum_{X_{T-p}=1}^{N} P(X_T, ..., X_{T-p}, X_t, ..., X_{t-p}|Y^{(T)}). \tag{3.35}$$

The total number of calculations required by the above algorithm is of order $N^{2(p+1)}T^2$ which is acceptable because usually N and p are fairly small.

Chapter 4
AR(1)HMM WITH APPLICATION TO TAO DATA

In this chapter we will focus on a bivariate autoregressive order one hidden Markov model (AR(1)HMM) with two states. Firstly we will present the model and discuss the empirical algorithms to recognize the state sequence and estimate the parameter set. Next we will use a set of simulated data to test the performance of the algorithm. Then we will apply the AR(1)HMM to an El Niño study by fitting the sea surface temperature data from Tropical Atmosphere Ocean Project (TAO) to the model. Moreover, a conventional HMM will also be built on the same data set and, through comparison, verify the strength of AR(1)HMM . At last, we will draw a conclusion on this study and further research on the subject are discussed.

4.1 Introduction of AR(1)HMM

4.1.1 Specifications of The Model

As the simplest case of multivariate autogressive hidden Markov models(MARHMM), one bivariate AR(1)HMM with two states could have the following form:

$$Y_t = \mu^{(X_t)} + \beta^{(X_t)}(Y_{t-1} - \mu^{(X_{t-1})}) + \epsilon_t \tag{4.1}$$

where Y_t is the bivariate observation vector in time t and $\mu^{(X_t)}$ is the mean vector

depending on the current state X_t :

$$Y_t = \begin{bmatrix} y_{t,1} \\ \\ y_{t,2} \end{bmatrix} \quad \text{and} \quad \mu^{(X_t)} = \begin{bmatrix} \mu_1^{(X_t)} \\ \\ \mu_2^{(X_t)} \end{bmatrix}.$$

$\beta^{(X_t)}$ is the autoregressive parameter of the current state X_t . It is a 2×2 diagonal matrix :

$$\beta^{(X_t)} = \begin{bmatrix} \beta_1^{(X_t)} & 0 \\ 0 & \beta_2^{(X_t)} \end{bmatrix}.$$

ϵ_t is the white noise with mean zero and covariance matrix Σ. It is independent to the current state.

$$\epsilon_t = \begin{bmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \end{bmatrix} \sim N(0, \Sigma) = N\left(0, \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}\right).$$

The parameter set of the model can be written as $\lambda = (\pi, A, B)$ for which:

1. π is the initial probability density matrix for first two states and

$$\pi = [\pi_{X_1 X_2}]_{2\times 2} = \begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix}.$$

2. A is the 2×2 transition matrix as usual: $A = [a_{ij}]_{2\times 2} = \begin{bmatrix} a_{11} & a_{12} \\ \\ a_{21} & a_{22} \end{bmatrix}$.

3. $B = (\mu, \Sigma, \beta)$ is the set of distribution parameter and autoregressive coefficients:

$$\mu = \begin{bmatrix} \mu^{(1)} & \mu^{(2)} \end{bmatrix} = \begin{bmatrix} \mu_1^{(1)} & \mu_1^{(2)} \\ \\ \mu_2^{(1)} & \mu_2^{(2)} \end{bmatrix} \quad , \quad \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

and

$$\beta^{(i)} = \begin{bmatrix} \beta_1^{(i)} & 0 \\ 0 & \beta_2^{(i)} \end{bmatrix}.$$

4.1.2 The Likelihood Function

Assume that all the parameters $\lambda = (\pi, A, B)$ are known. With the model structure described above, we have

$$\epsilon_t = Y_t - \mu^{(X_t)} - \beta^{(X_t)}(Y_{t-1} - \mu^{(X_{t-1})}). \tag{4.2}$$

Since $\epsilon_t \sim N(0, \sum)$, is independent of t, the Jacobian of the transformation from ϵ_t to Y_t does not depend on t and it is equal to $\left|\Sigma^{-\frac{1}{2}}\right|$. Now using this Jacobian, we can write the joint density of $Y_1, Y_2, ..., Y_T$ as:

$$f(Y_1, Y_2, ..., Y_T|\ i_1, i_2, ..., i_T) = (2\pi)^{-\frac{T}{2}} \left|\Sigma^{-\frac{T}{2}}\right| \exp\left\{-\frac{1}{2}\sum_{t=1}^{T} \epsilon_t' \Sigma^{-1} \epsilon_t\right\} \tag{4.3}$$

when the state sequence $X_1 = i_1$, $X_2 = i_2, ...,$ $X_T = i_T$ are given.

Hence

$$f(Y_1, Y_2, ..., Y_T,\ i_1, i_2, ..., i_T) = \pi_{i_1} a_{i_1 i_2} ... a_{i_{T-1} i_T} (2\pi)^{-\frac{T}{2}} \left|\Sigma^{-\frac{T}{2}}\right| \exp\left\{-\frac{1}{2}\sum_{t=1}^{T} \epsilon_t' \Sigma^{-1} \epsilon_t\right\} \tag{4.4}$$

and the joint density function of $Y_1, Y_2, ..., Y_T$ is given by

$$f(Y_1, Y_2, ..., Y_T) = \sum_{All\ I} f(Y_1, Y_2, ..., Y_T,\ i_1, i_2, ..., i_T) \tag{4.5}$$

where $I = \{i_1, i_2, \cdots, i_T\}$ is any one of the state paths.

Equation (4.5) indicates that the likelihood of any realization of observations $Y_1, Y_2, ..., Y_T$ can be calculated through summing up the expression (4.4) for all the possible state sequences.

Apparently this straight-forward method is computationally intractable when long state sequences are involved. Now we have almost the same problem as in Section 1.5 when dealing with the likelihood function for conventional HMM. Though the autoregressive structure of AR(1)HMM makes the situation more complicated, a modified forward-method could solve the problem efficiently.

For AR(1)HMM, let's define the new forward variable $\phi_t(X_{t-1}, X_t) = f(Y_1, Y_2, ..., Y_t, X_{t-1}, X_t)$, the joint density function of partial observations up to time t and the most recent two states. Then it's not hard to see:

$$\begin{aligned}\phi_{t+1}(X_t, X_{t+1}) &= f(Y_1, Y_2, ..., Y_t, X_t, X_{t+1}) \\ &= \sum_{X_{t-1}} f(Y_1, Y_2, ..., Y_t, X_{t-1}, X_t) a_{X_t X_{t+1}} f(Y_{t+1} | Y_t, ..., Y_1, X_{t-1}, X_t, X_{t+1}).\end{aligned}$$

Then joint density function $f(Y_1, Y_2, ..., Y_T)$ can be calculated with the following iterative procedures:

- Step 1: Initialization

$$\phi_2(X_1, X_2) = \pi_{X_1 X_2} (2\pi)^{-\frac{1}{2}} \left|\Sigma^{-\frac{1}{2}}\right| \exp\left\{-\frac{1}{2}\epsilon_2' \Sigma^{-1} \epsilon_2\right\}. \tag{4.6}$$

- Step 2: For $t = 2$ to $T - 1$,

$$\phi_{t+1}(X_t, X_{t+1}) = \sum_{X_{t-1}} \phi_t(X_{t-1}, X_t) a_{X_t X_{t+1}} (2\pi)^{-\frac{1}{2}} \left|\Sigma^{-\frac{1}{2}}\right| \exp\left\{-\frac{1}{2}\epsilon'_{t+1}\Sigma^{-1}\epsilon_{t+1}\right\}. \tag{4.7}$$

- Step 3: Finish up

$$f(Y_1, Y_2, ..., Y_T) = \sum_{X_T} \sum_{X_{T-1}} \phi_T(X_{T-1}, X_T). \tag{4.8}$$

When the likelihood function of AR(p)HMM ($p > 1$) is studied, almost the same procedure can be employed with the definition of forward variable change to $\phi_t(X_{t-p}, ..., X_t) = f(Y_1, Y_2, ..., Y_t, X_{t-p}, ..., X_t)$.

4.1.3 Scaling Technique

When the observation sequence is fairly long (Approximately>50), the value of likelihood function will become extremely small that goes beyond the computational precision of any computer system. So a scaling procedure for the calculation of likelihood function is necessary. The idea of scaling procedure is to multiply the forward variable $\phi_t(X_{t-1}, X_t)$ by a factor independent of the states X_{t-1} and X_t . One good choice is to divide $\phi_t(X_{t-1}, X_t)$ by its sum over all states:

$$\phi^*_t(X_{t-1}, X_t) = \frac{\phi_t(X_{t-1}, X_t)}{\sum_{X_{t-1}} \sum_{X_t} \phi_t(X_{t-1}, X_t)} \tag{4.9}$$

where $\phi_t^*(X_{t-1}, X_t)$ is the scaled forward variable.

If using the scaled forward variables $\phi_t^*(X_{t-1}, X_t)$ all along the calculation, we know the value of likelihood function (4.8) will be 1 no matter what the observations are. The real value of the likelihood function would just be the products of all scaling denominators. Or one could get the log-likelihood function by summing up all the log form of them:

$$L = \log f(Y_1, Y_2, ..., Y_T) = \sum_{t=2}^{T} \log \left(\sum_{X_{t-1}=1}^{2} \sum_{X_t=1_t}^{2} \phi_t(X_{t-1}, X_t) \right). \tag{4.10}$$

4.1.4 Initialization Problem

The estimation of AR(1)HMM parameters will use the segmental K-mean algorithm. As described in Section 1.8, the segmental K-mean algorithm is an iterative procedure and the parameter set must be initialized before the iterations start.

As mentioned in the Section 1.7, either E-M algorithm or segmental K-mean algorithms could only lead to a local maximum of the HMM likelihood function. For AR(1)HMM, this is also true. To get the parameter estimates with a global maximum likelihood, a grid search approach[20] might be used. In grid search approach, the parameter space is seen as a grid with many small cells and all the vertices are used as the initial values of the parameters. Because the parameter space is so big in the case of AR(1)HMM, the grid search method requires considerable computational power which is intractable for practical purposes. So in this study, we just use the simple

method which initializes the parameters using a rough estimation of the state path. The method will be described in the next section. Please note that our initialization of the parameters will possibly only lead to a local maximum.

4.2 Model Estimation

The method we used to estimate the model parameters are a modified version of conventional Segmental K-mean Algorithm (SKA) . A little more detailed description of SKA have already been introduced in Section 1.8. So here we will focus on the procedures of the algorithm.

The estimation can be achieved by following iterative steps:

- **Step 1**: Initialization.

  Firstly one need to initialize the unknown state sequence by clustering all the observations into several state groups . That means, if an observation Y_t is grouped into a state group i, we assume the corresponding state X_t be the ith state. In case of only two possible states , we could simply assign each observation to a state by comparing its norm (Euclidean distance to origin) to a threshold. Those whose norm are greater than the threshold are assume to be in state 1 and rest are assume to be in state 2. The choice of the threshold can be the average of all the norms, or simply by guess through the visualization of

the data.

Once we have a initial state path $\mathbf{X}^* = \{X_1^*, X_2^*, \cdots, X_T^*\}$, we could initialize the parameter set $\lambda = (\pi, A, B)$ by the following estimators:

(1) The transition matrix A can be initialized by:

$$\hat{a}_{ij} = \frac{Number\ of\ transitions\ from\ state\ i\ to\ state\ j}{Number\ of\ transitions\ from\ state\ i}. \tag{4.11}$$

(2) The the initial probabilities π can be set to be equal to transition matrix A.

(3) $\mu_j^{(i)}$, the jth element of mean vector in state i :

$$\hat{\mu}_j^{(i)} = \frac{\sum_{X_t=i} y_{t,j}^{(X_t)}}{N_i}. \tag{4.12}$$

N_i is the number of states i in the whole state sequence.

(4) $\beta_j^{(i)}$, the jth element autoregressive parameters in state i :

$$\hat{\beta}_j^{(i)} = \frac{\sum_{X_t=i}(y_{t,j} - \hat{\mu}_j^{(X_t)})(y_{t-1,j} - \hat{\mu}_j^{(X_{t-1})})}{\sum_{X_t=i}(y_{t,j} - \hat{\mu}_j^{(X_t)})^2}.. \tag{4.13}$$

(5) Σ , the covariance matrix of the white noise:

$$\hat{\Sigma} = \frac{\sum_{t=1}^{T} \hat{\epsilon}_t' \hat{\epsilon}_t}{T}. \tag{4.14}$$

where $\hat{\epsilon}_t = Y_t - \hat{\mu}^{(X_t)} - \hat{\beta}^{(X_t)}(Y_{t-1} - \hat{\mu}^{(X_{t-1})})$ as defined before.

- **Step 2**: Calculate the smoothed probability $P(X_t, X_{t-1}|Y)$ and $P(X_t|Y)$ based on the estimated parameter set $\hat{\lambda} = (\hat{A}, \hat{B}, \hat{\pi})$ as described in Section 3.5.

- **Step 3**: Find the new optimal state sequence X^* based on the value of $P(X_t|Y)$:

  If $P(X_t = 1|Y) > P(X_t = 2|Y)$

  Then $X_t^* = 1$

  Else $X_t^* = 2$.

- **Step 4**: Re-estimate the parameters $\lambda = (\pi, A, B)$ based on new optimal state path X^* :

  (1)

$$\hat{a}_{ij} = \frac{\sum_{t=2}^{T} P(X_t = j, X_{t-1} = i|Y)}{\sum_{t=2}^{T} P(X_{t-1} = i|Y)}. \tag{4.15}$$

  (2)

$$\pi_{ij} = P(X_2 = j, X_1 = i|Y). \tag{4.16}$$

  (3) $\mu_j^{(i)}$,$\beta_j^{(i)}$ and Σ would be estimated through equation 4.12 to 4.14.

- **Step 5**: If there are any chance in state path X^* , repeat step 2 to step 5.

It has been proven in [18] that the procedures above would lead to the convergence of target state optimized likelihood function. We will evaluate the performance of above algorithm with the test data in the next section.

4.3 Model Testing

A bivariate observation sequence of length l000 has been simulated from the model (equation 4.1) described in Section 4.1. The values of parameters used to simulate the data are listed in the column "Original Parameters" of Table (4.1). The total number of parameters of real value is 20. Moreover, there are noticeable difference in the mean μ's and autoregressive parameters β's between the two state to ensure the the feasibility of state recognition.

Figure 4.1 shows a 2-D plot of the first element against and second element ($y_{t,1}$ vs $y_{t,2}$) of the first 100 observations. It's obvious the points are gathering into two clusters , but their boundary is somewhat blurry. Figure 4.2 shows the time series plot for $y_{t,1}$ and $y_{t,2}$, individually.

The estimation procedure are described in last section and repeated three times for the first 100, first 300 and whole 1000 data. The results are reported in the Table 4.1.

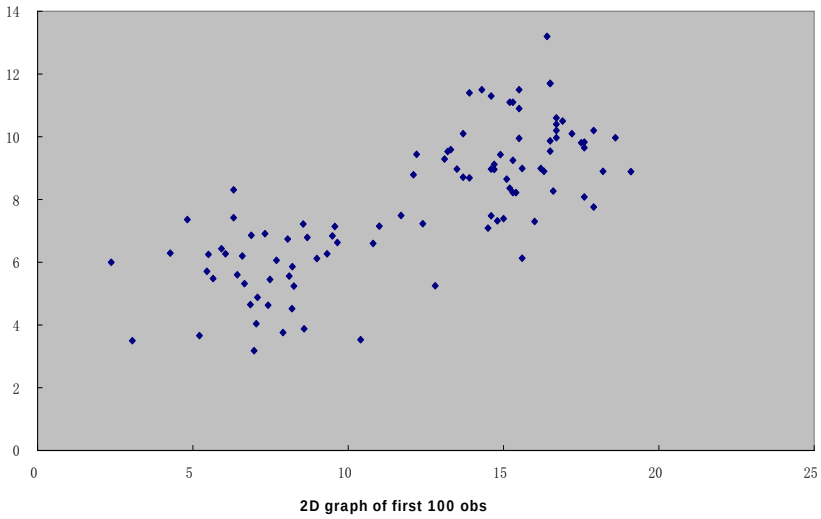

Figure 4.1 2-D Graph of First 100 Observations

Since there are only one sequence involved, the estimation of initial density matrix of π would have 1 in one entry and 0's in the rest entries. In fact , in most applications , initial probability density would not have any realistic meanings because the time series data , in most cases, don't have a beginning point.

The last row $logL$ in the table is the log-likelihood value $log(P(\mathbf{Y}|\hat{\lambda}))$ computed from the estimated parameters and the whole 1000 observations. It is listed here as a measure to compare the goodness-of-estimation for different data set. The calculation of $logL$ follows the forward-procedure in Section 4.1.2.

In the above example, as the size of test data increases from 100, 300 to 1000, the log-likelihood value, $logL$, has also increased from -6174.8, -5700.9 to -5236.6. It

is true that the estimates improve as the data set for the training model increase. For example, the $logL$ value archived by Training set 3 (-5236.6) are pretty close to the real one (-5207.1) which indicates a very good estimation. But it should be pointed out that the Segmental K-mean Algorithm are based on the maximum state optimized criterion, namely to maximize the $L(\lambda|\mathbf{Y}, \mathbf{X}^*)$ rather than $L(\lambda|\mathbf{Y})$. So it would be no surprise that sometimes the estimation with the shorter observation sequence has a greater value of $logL$ than the longer ones. But as a matter of fact, the estimations for ARHMM based on the maximum state optimized criterion and maximum likelihood

	Original Parameters	Test set 1(100 data)	Test set 2(300 data)	Test set 3(1000 data)
A	$\begin{bmatrix} 0.85 & 0.15 \\ 0.3 & 0.7 \end{bmatrix}$	$\begin{bmatrix} 0.8937 & 0.1063 \\ 0.1695 & 0.8305 \end{bmatrix}$	$\begin{bmatrix} 0.8316 & 0.1684 \\ 0.2784 & 0.7216 \end{bmatrix}$	$\begin{bmatrix} 0.8447 & 0.1553 \\ 0.2715 & 0.7285 \end{bmatrix}$
π	$\begin{bmatrix} 0.7 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 1.0 \end{bmatrix}$
$\mu^{(1)}$	$\begin{bmatrix} 15.2 \\ 9.3 \end{bmatrix}$	$\begin{bmatrix} 15.4663 \\ 9.3407 \end{bmatrix}$	$\begin{bmatrix} 15.2101 \\ 9.4109 \end{bmatrix}$	$\begin{bmatrix} 15.2487 \\ 9.3355 \end{bmatrix}$
$\mu^{(2)}$	$\begin{bmatrix} 7.4 \\ 5.4 \end{bmatrix}$	$\begin{bmatrix} 7.4002 \\ 5.8054 \end{bmatrix}$	$\begin{bmatrix} 7.3666 \\ 5.5270 \end{bmatrix}$	$\begin{bmatrix} 7.2351 \\ 5.4436 \end{bmatrix}$
$\beta^{(1)}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & -0.3 \end{bmatrix}$	$\begin{bmatrix} 0.0538 & 0 \\ 0 & -0.3001 \end{bmatrix}$	$\begin{bmatrix} 0.1765 & 0 \\ 0 & -0.2871 \end{bmatrix}$	$\begin{bmatrix} 0.1469 & 0 \\ 0 & -0.2731 \end{bmatrix}$
$\beta^{(2)}$	$\begin{bmatrix} -0.7 & 0 \\ 0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} -0.3863 & 0 \\ 0 & 0.5006 \end{bmatrix}$	$\begin{bmatrix} -0.3668 & 0 \\ 0 & 0.5842 \end{bmatrix}$	$\begin{bmatrix} -0.5098 & 0 \\ 0 & 0.5187 \end{bmatrix}$
Σ	$\begin{bmatrix} 3.3 & 0.3 \\ 0.3 & 2.4 \end{bmatrix}$	$\begin{bmatrix} 3.0161 & 0.3257 \\ 0.3257 & 1.6737 \end{bmatrix}$	$\begin{bmatrix} 3.2163 & -0.0741 \\ -0.0741 & 2.2021 \end{bmatrix}$	$\begin{bmatrix} 3.4496 & 0.1086 \\ 0.1086 & 2.4852 \end{bmatrix}$
$\log L$	-5207.1	-6174.8	-5700.9	-5236.6

Table 4.1 Summary of Test Result

criterion would be consistent with each other under suitable conditions, or in other words , they will be very close.

A byproduct of the algorithm is a set of smoothed probability $P(X_t = i|\mathbf{Y}, \hat{\lambda})$, the probability (or likelihood) of state i at time t based on the whole observation sequence and estimated parameter set. In the Segmental K-mean algorithm, these smoothed probabilities have been used to draw inferences on the state process. A graph of smoothed probability $P(X_t = 1|Y, \hat{\lambda})$ based on the parameter estimates from training set 3 for the first 100 observation vectors has been reported in Figure 4.3. For non-autoregressive univariate HMM, a proper estimation will lead to the smoothed probability curve very similar to the time series plot of observation sequence. For autogressive ones, this relation would also exist but somewhat less obviously.

The computer programs used in the above and following applications are designed by the author and named AR1HMM. AR1HMM includes a set of MATLAB functions that implement the Segmental K-mean Algorithm to estimate the model parameters of ARHMM of order 1. Now it is only designed to handled the univariate or bivariate AR(1) hidden Markov Model with two states. But it would be easy to extend the programs to deal with more general AR(p) cases with more possible hidden states. For a detailed description of AR1HMM , see Appendix I.

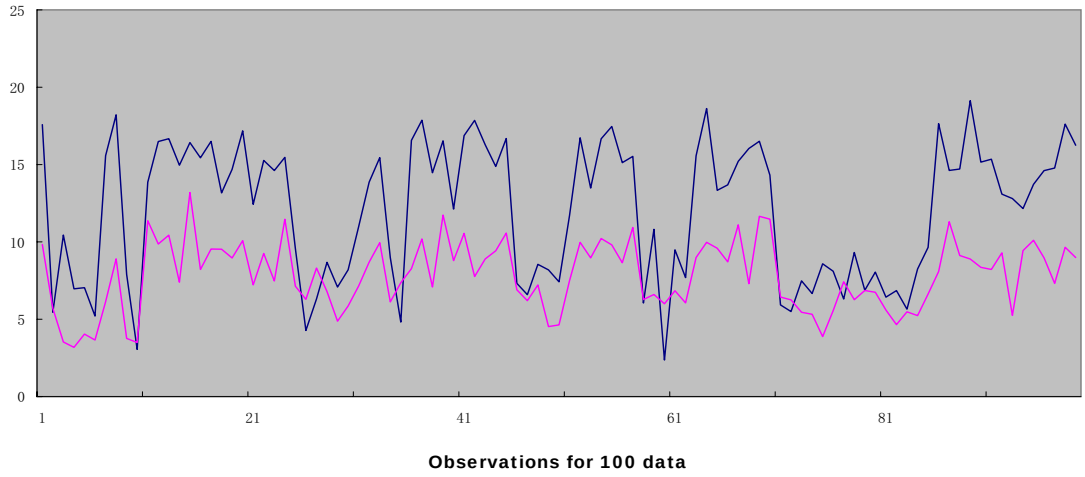

Figure 4.2 Time Series Plot for $y_{t,1}$ and $y_{t,2}$

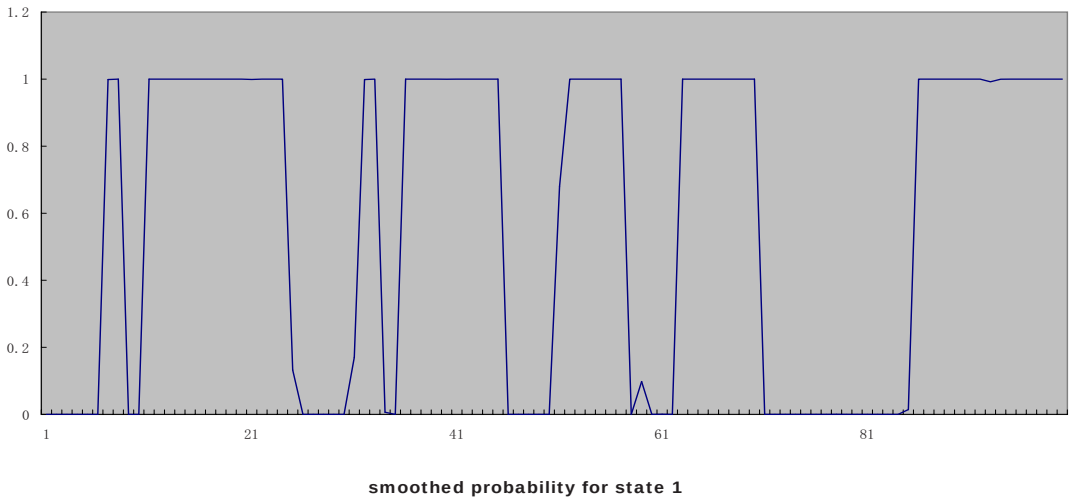

Figure 4.3 Time Series Plot for smoothed probability $P(X_t = 1|Y, \hat{\lambda})$

4.4 Application to TAO data

In this section , we will use AR(1)HMM to model the sea surface temperature data from the Tropical Atmosphere Ocean (TAO) Project. Also a conventional hidden Markov model will also be built on the same data as a comparison.

4.4.1 Overview and Data Preparation

El Niño is a disruption of the ocean-atmosphere system in the tropical Pacific having important consequences for weather around the globe. El Niño as a physical phenomenon is a proven fact. But the way it works has many theories. An autoregressive hidden Markov model could interpret the frequent El Nino phenomena as the results of stochastic switches between two unobserved states (normal state and abnormal state). Under this assumption, we could make statistical inferences on El Nino based on the available observations such as sea surface temperatures whose changes have been regarded as an important measure of the occurrence of El Nino.

Between 1985 and 1994, the TAO Project Office of Pacific Marine Environmental Laboratory (PMEL) instrument the entire tropical Pacific with nearly 70 ocean moorings which enables the real-time collection of high quality oceanographic and surface meteorological data for monitoring, forecasting, and understanding of climate swings associated with El Niño and La Niña. All the data can be downloaded free from PMEL's website (http://www.pmel.noaa.gov/tao/).

Among all the data, sea surface temperature (SST) is of most importance because

it is the most direct measure for the occurrence the El Niño phenomena. In TAO, all the SST data are collected from buoys placed in the water of 1-meter depth. Figure 4.4 shows the distribution of all buoys in the Pacific Ocean. Two red points in the middle represents the positions (0N170W and 0N155W, representing 0 on the y-axis and 170 and 155 on the x-axis respectively) where the data are going to be studied in this section. Namely, the SST data from these two positions, in the form of time series, are going to be the first and second entries of observation vector $Y_t = [y_{1,t}, y_{2,t}]'$.

The reasons these two sites are selected are:

1. They are both in the equator, so there will be no seasonal effect in the model.

2. The SST data available at these two sites (showed in Figure 4.5) are relatively complete (Although there are still some missing value) and largely overlapping which can be easily turn into a bivariate observation sequence.

3. They are relatively close, so it's relatively more likely they will be in the same state at same time.

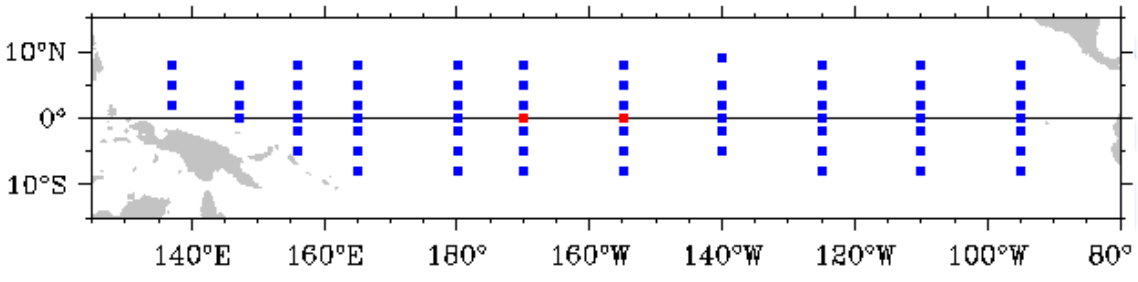

Figure 4.4 Buoy Distribution and Selection

All the SST data obtained from the TAO website are in Celsius scale, with two places of decimal. The original data from 0N170W (sequence 1) is the daily average temperature from 16, May 1988 to 5 Apr 2004. It is in 5 blocks and totally 5804 values , with 56 missing values. The original data from 0N155W (sequence 2) is collected from 21, Jul 1991 to 4, Apr 2004. It has 1 block and totally 4642 values and 33 missing values.

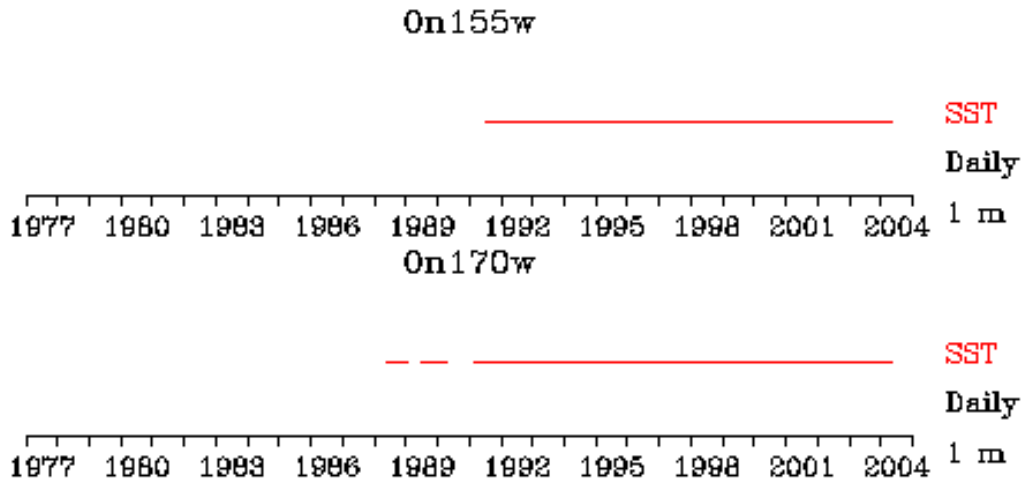

Figure 4.5 Data Availability in Two Sites

To form the bivariate observation sequence, the sequence 1 are chopped into the same length as sequence 2.

Since all the gaps between the blocks in sequence 1 are quite small (maximal length of 6) , they can be seen as another sort of missing values. There are many ways to deal with the missing values. Here we just "smooth" the missing values with

the average of their most closed neighbors. In total, the data have been modified are below 2% of total.

4.4.2 Model Estimation

Now we use the AR(1)HMM described in the previous sections to model the SST data. Hopefully the estimation can be used to interpret the recent El Niño.

Figure 4.6 and Figure 4.7 show two identical time series plots of observation $y_{1,t}$, $y_{2,t}$. The blue curve in the graph represents the $y_{1,t}$ (the SST in 0N170W) and the green one represents the $y_{2,t}$ (the SST in 0N155W).

We use $[28.5, 27]'$ as the threshold value for the initial guess. Namely, those observations $Y_t = [y_{1,t}, y_{2,t}]'$ whose norms are greater than $\sqrt{28.5^2 + 27^2}$ will assign to state 1 (El Niño state) and the rest will assign to state 2 (normal state) in the initialization of algorithm. 28.5 and 27 are rounded averages for the whole SST observation sequence at the two sites. Once the state sequence are initialized, the parameter set can be easily initialized following the procedure of Section 4.2.

The computer program used in the estimation is still AR1HMM. The program running the MATLAB V6.2 on a Pentium 4 2.0GHZ PC with 512M RAM, took around 6 hours for convergence.

To compare the competence between ARHMM and conventional HMM, a two-state HMM with bivariate Gaussian state-conditional distribution are also built for the

same data set. For an specification of an conventional HMM, please refer to chapter one. The initialization procedure for the HMM are almost the same as AR(1)HMM, except for the obvious fact that there are no autoregressive coefficients in the model.

The HMM estimation is based on the EM algorithms and are implemented by H2M. The running time for the estimation on the same computer is only 7 minutes.

4.4.3 Results and Analysis

Table 4.2 reports the estimation of parameters for both AR(1)HMM and conventional HMM. The bold red straight lines in the bottom and top of Figure 4.6 and Figure 4.7 represents the estimated state path of State 1 and State 2 respectively.

The first row of table 4.2 compares the two estimated transition matrix between AR(1)HMM and HMM. For both matrices, the estimated transition matrix A has very high values in the diagonal entries while the off-diagonal entries are trivial. This feature indicates there would be no frequent jumps between two states for both models. Once it is in a state (either in normal state or in abnormal state), it will stay here for quite a long time before the next jump. In practice, El Niño phenomena is not a daily weather changes like rains and fogs. It is a disruption of the ocean-atmosphere system and its formation and dissolution takes months. The high diagonal values of the transition matrices for both models ensure the stability of state estimates. Comparatively, the diagonal elements of AR(1)HMM is a little bit greater than the

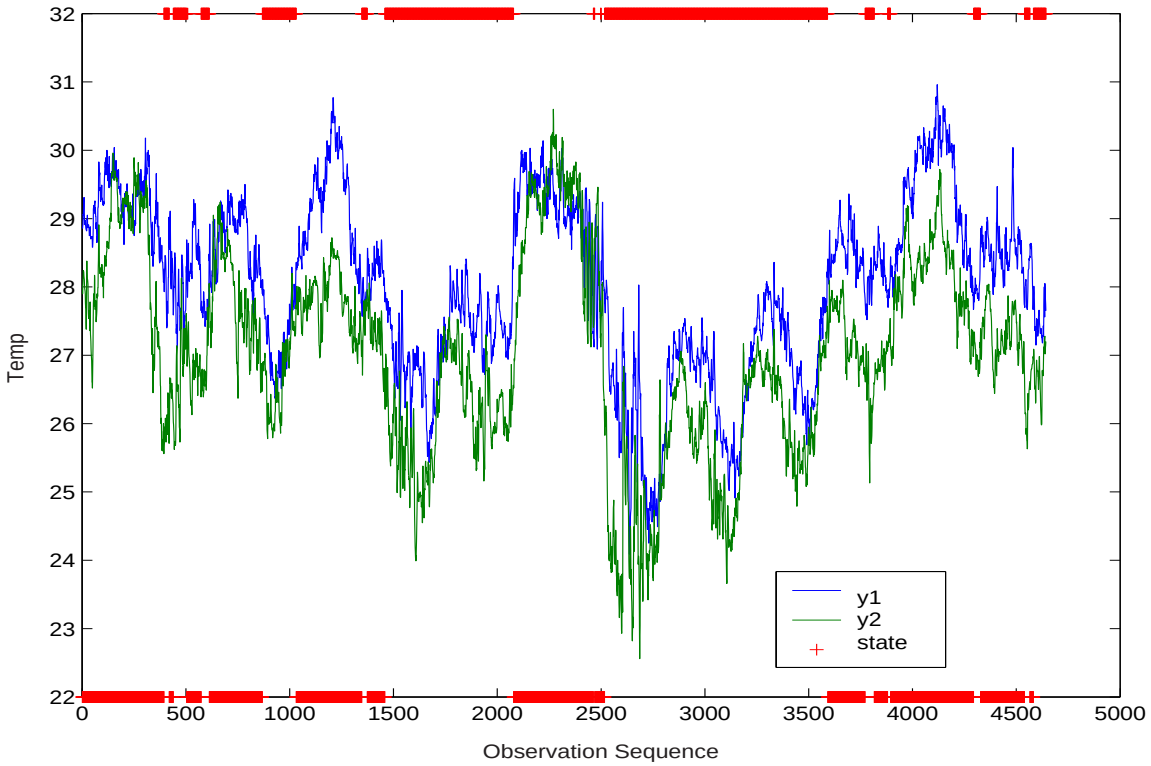

Figure 4.6 Observations and the HMM estimated state path

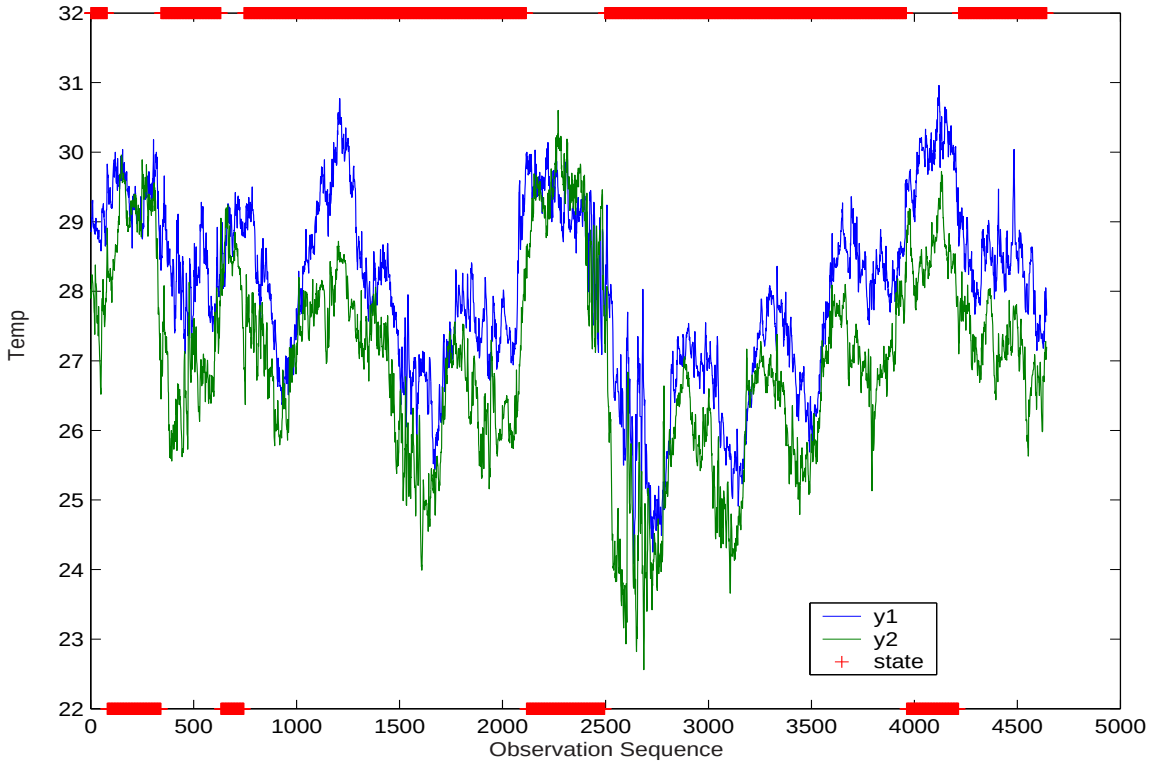

Figure 4.7 Observations and the AR1HMM estimated state path

ones for HMM. Correspondingly, a comparison of their respective state paths in Figure 4.7 (AR(1)HMM) and Figure 4.6 (HMM) reveals that the state path of AR(1)HMM are more stable and more coincident to the reality.

In the second row of Table 4.2, the stationary probability distribution of transition matrices for both models are worked out as $\overline{A}$. The stationary probability distribution is important parameters for the Markov chain. The elements in the stationary probability distribution vector correspond to the long-run frequencies of the occurrence of each state. The two elements of $\overline{A}$ for HMM is 0.5047 and 0.4953. It indicates that by HMM prediction, the total time of being in the abnormal state and being in the normal state would be approximately same in the long run. But this is not true, at least for the weather situation in the last few years. Relatively, the AR(1)HMM are close to reality. It has a stationary probability distribution [0.2157 0.7843] which indicates only approximately one fifth of the time in the last few years were we in the El Niño state.

The sea surface temperature (SST) is one of the most direct measure of the occurrence of El Niño. Hence the most important distinction between the normal state and El Niño state lies in their difference in the mean SST values. Referred to the 4th and 5th rows of Table 4.2. The average SST of the El Niño state is 1.6C and 2.3C higher than the normal state in the two sites respectively by AR(1)HMM estimates. Based on our knowledge of El Niño, this difference in Centigrade should be a little greater

	Estimated Parameters by AR(1)HMM	Estimated Parameters by HMM
A	$\begin{bmatrix} 0.9960 & 0.0040 \\ 0.0011 & 0.9989 \end{bmatrix}$	$\begin{bmatrix} 0.9947 & 0.0053 \\ 0.0054 & 0.9946 \end{bmatrix}$
$\overline{A}$	$\begin{bmatrix} 0.2157 & 0.7843 \end{bmatrix}$	$\begin{bmatrix} 0.5047 & 0.4953 \end{bmatrix}$
π	$\begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 \end{bmatrix}$
$\mu^{(1)}$	$\begin{bmatrix} 29.3821 \\ 28.8702 \end{bmatrix}$	$\begin{bmatrix} 28.9935 \\ 28.0240 \end{bmatrix}$
$\mu^{(2)}$	$\begin{bmatrix} 27.6562 \\ 26.5284 \end{bmatrix}$	$\begin{bmatrix} 26.9738 \\ 25.9526 \end{bmatrix}$
$\beta^{(1)}$	$\begin{bmatrix} 0.9640 & 0 \\ 0 & 0.9564 \end{bmatrix}$	N/A
$\beta^{(2)}$	$\begin{bmatrix} 0.9905 & 0 \\ 0 & 0.9873 \end{bmatrix}$	N/A
Σ	$\begin{bmatrix} 0.0248 & 0.0076 \\ 0.0076 & 0.0032 \end{bmatrix}$	$\Sigma^{(1)} = \begin{bmatrix} 0.4292 & 0.3528 \\ 0.3528 & 0.8349 \end{bmatrix}$
		$\Sigma^{(2)} = \begin{bmatrix} 0.7359 & 0.5932 \\ 0.5932 & 0.9740 \end{bmatrix}$

Table 4.2 Summary of Parameter Estimation

in reality (usually $> 3\,^{\circ}$C). We will see later this is mostly because the AR(1)HMM takes sea surface temperature variations caused by the unusual warm ocean current in 1994 as random errors, rather than an occurrence of El Niño. Or in other word, due to its relative simplicity, this AR(1)HMM has a more critical standard to admit an El Niño than meteorologists' standard. In a similar manner, the estimates of HMM are a little lower for both states, but the difference is approximately the same.

Another notable feature of AR(1)HMM estimation is the high autoregressive coefficients. This is exactly the reason why AR(1)HMM are superior than conventional HMM in this application. Conventional HMM assumes there are independency relation between the observations. But this is rarely the case for time series observations. As in this application, SST data are collected on a day-by-day bases and apparently the independency assumption is inappropriate. Comparatively , the autoregressive structure contributes the superiority of AR(1)HMM in a way it prevents the frequent fluctuations of state path. This difference are clearly shown in the comparison of Figure 4.6 and Figure 4.7. Conventional HMM (Figure 4.6) are very sensitive to the numerical swings of the current SST and hence mistakes several fluctuations of SST as the switches of states. While for the same data, AR(1)HMM state path are more stable and close to reality.

Very often El Niño are visualized by a graph of Sea Surface Temperature in Pacific Ocean. Figure 4.9 is downloaded from TAO website and shows the mean and anoma-

lies of sea surface temperature from 1986 to 2004. Time is increasing downwards from 1986 at the top of the plot, to 2004 in the bottom of plot. The red color on the left is the warm pool of water typically observed in the western Pacific Ocean. The warm tongue from western varies seasonally. El Niño is characterized by the exaggeration of the usual seasonal cycle. For example in 1997-1998 there is a warm water (red) penetrating eastward and that is a strong El Niño. From the graph, it is not difficult to see there are El Niños 1986-1987, 1991-1992, 1993, 1994 and 1997.

The red lines uprightly placed in the left and right of the graph were added by the author indicating the AR(1)HMM estimated state path. We can see that the state path of El Niño (on the right) estimated from this AR(1)HMM are well consistent with the real occurrences of El Niño in recent years except for 1994. In 1994, we can see from the graph the warm tongue stick up to east which indicates an occurrence of El Niño. But the AR(1)HMM does not admit it as an El Niño state.

Table 4.3 corresponds the estimated states of AR(1)HMM with their accurate date intervals. It seems the estimation only misses the El Niño in 1994 because it is less severe than others.

Fig 4.8 is the same graph with HMM estimated state path. Again we have the conclusion that AR(1)HMM has a more robust estimation due to its autoregressive structure.

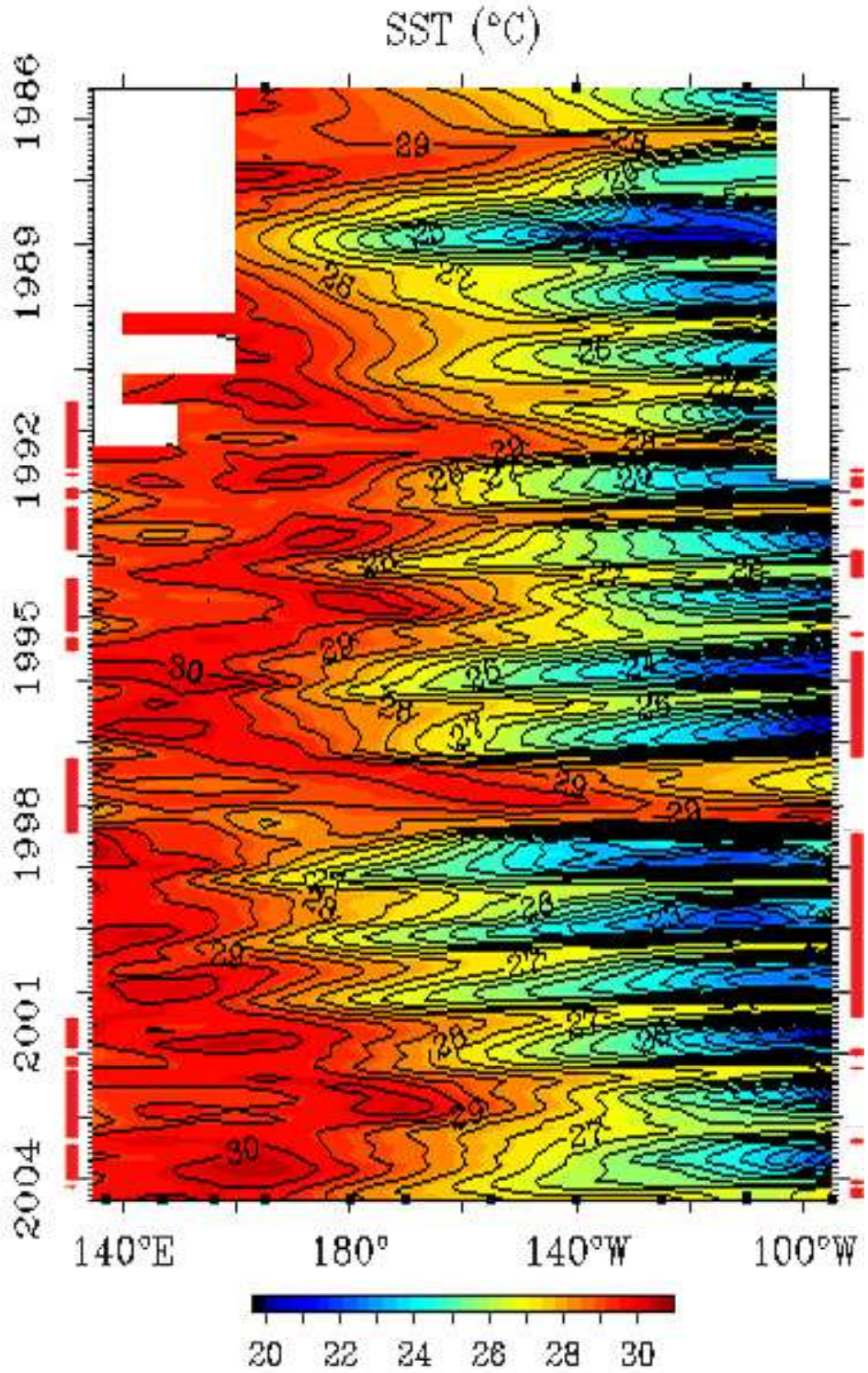

Figure 4.8 Mean and anomalies of SST with HMM estimated states 1986-2004

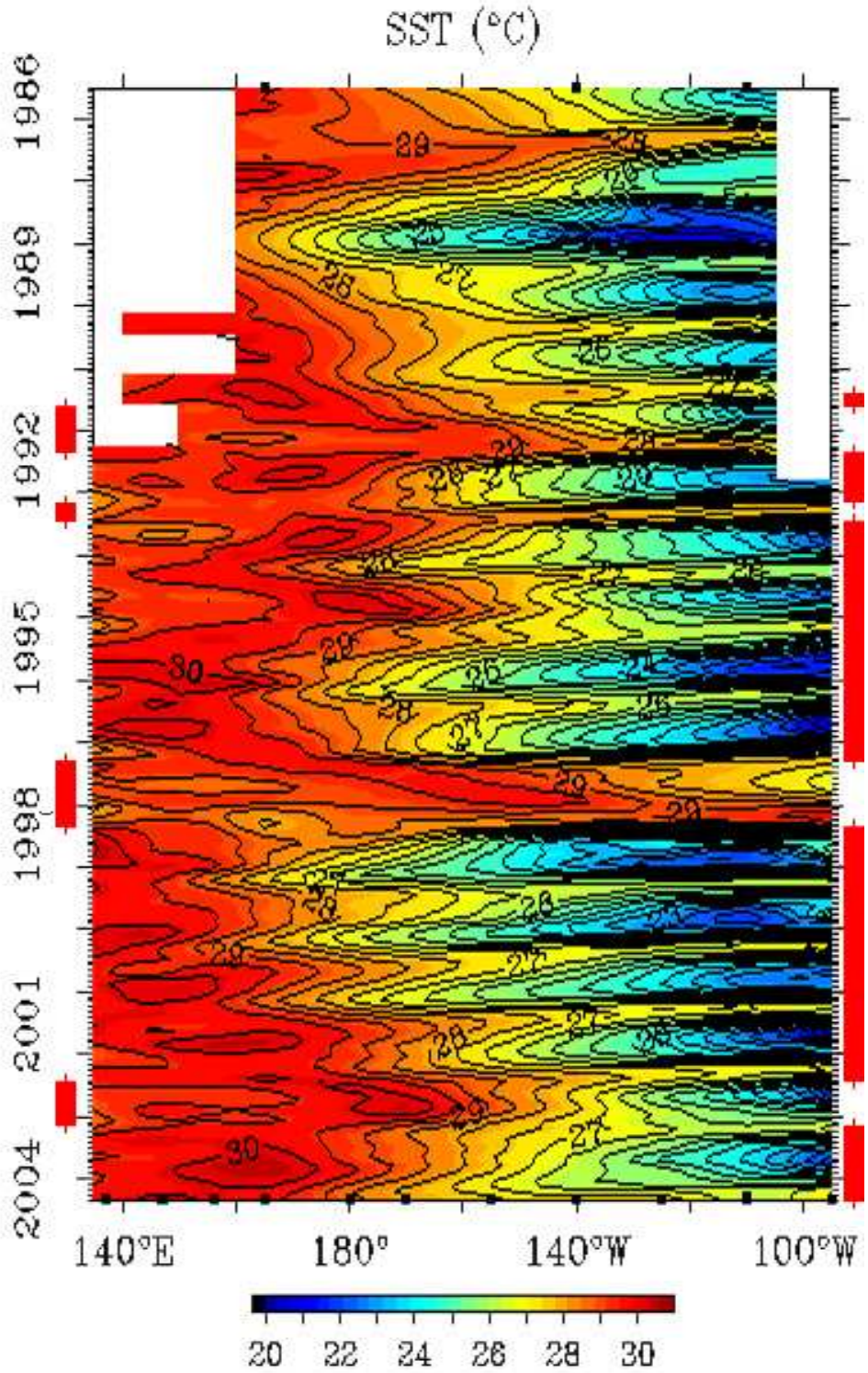

Figure 4.9 Mean and anomalies of SST with AR1HMM estimated states 1986-2004

Period	State	Period	State
1991,07,21-1991,10,08	2	1997,05,05-1998,05,20	1
1991,10,08-1992,06,25	1	1998,05,21-2002,05,24	2
1992,06,26-1993,04,12	2	2002,05,25-2003,02,01	1
1993,04,13-1993,08,02	1	2003,02,02-2004,04,04	2
1993,08,03-1997,05,04	2		

Table 4.3 State Path by Date

4.5 Conclusion

Hidden Markov model has already gained its popularity in many applications including climatology [29] due to its flexible and rich mathematical structure. But when it comes to the chronological data, the independency assumption of HMM are not appropriate any more because the correlation structure is essentially a nature of time series. Autoregressive hidden Markov model is a combination of time series and hidden Markov chains. It is more mathematical rich. It is more applicable. It is more complex. J.D.Hamilton has developed an estimation algorithm to solve the univariate ARHMM based on the E-M algorithm [9]. The ARHMM estimation of L.R.Rabiner could deal with multivariate ARHMM but it is not mathematical sound due to the usage of a number of unconvincing numerical approximations. In this thesis, an estimation of a simple multivariate ARHMM (with autoregressive structure of order

1 and only 2 possible states) based on the segmental K-mean algorithm has been developed. Also it might be the first time that the ARHMM has been engaged in the El Niño study.

Considering only two sites (0N170W and 0N155W) with limited data have been used in the study, the performance of the AR(1)HMM is very well. AR(1)HMM has successfully recognized all but one El Niños in the recent years. Furthermore, the comparisons with conventional HMM have confirmed the strength of autoregressive structure in stabilizing the state fluctuations.

4.6 Proposal for Future Research

In this chapter we have modelled the sea surface temperature data with an autoregressive hidden Markov model (with only two possible states) to predict the occurrences of El Niños. There are many directions in which the model could be extended.

First of all, the inclusion of a third state, "La Niña state" could be both convenient and beneficial. La Niña is the twin phenomena of El Niño, characterized by unusual cold ocean temperatures in Equatorial Pacific. The inclusion of the La Niña state would lead to a better understanding of the ocean-atmosphere system without making extensive changes to the current model and is likely to improve the accuracy of recognition of both El Niño and normal states. An even further extension of the model is to fractionize the El Niño state and La Niña state into several sub-states.

Z.Psaradakis and N.Spagnolo[26] suggest several procedures to determine the number of states required for the ARHMM to adequately characterize the observations.

Secondly, more data should be used in the future of El Niño research. In this study, only sea temperature data from two sites are used to train the model. More sea temperature data, as well as some other indicators for El Niño should be used as high dimensional observation vectors in the model. High dimensional observations would inevitably lead to a more complex covariance structure of the model and to solve this, some spatial techniques might be used in the study. Also larger amount of data may require a greater computational capacity, but will lead to more accurate and more interesting results.

Thirdly, one could improve the parameter estimation algorithm in our ARHMM study in several ways. Firstly, one might be able to accelerate the convergence speed by estimating the initial probability π by $\overline{A}$, the stationary distribution of transition matrix in the re-estimation formula. Also, one could employ more sophisticated approaches in the initialization procedure such as the complexity-penalized likelihood maximization approach employed in [26].

Finally, one could use different forms of autoregressive hidden Markov models. As mentioned in Chapter 3, there are several forms of ARHMM. For instance, a form of ARHMM may have the state dependent covariance matrix Σ_{X_t} for its error term u_t. This model would be more complex but the results might be rewarding.

Appendix A
AR1HMM : MATLAB functions for the estimation of autogressive hidden Markov model.

A.1 Introduction

AR1HMM is a set of MATLAB functions that implement a modified version of segmental K-mean algorithm to estimate the parameter set of an autogressive hidden Markov model in the case of bivariate Gaussian state-depended distribution, autogressive order one and two hidden states.

While autogressive hidden Markov models have several different forms, the programs are especially designed to deal with one of them, namely formula (4.1)

$$Y_t = \mu^{(X_t)} + \beta^{(X_t)}(Y_{t-1} - \mu^{(X_{t-1})}) + \epsilon_t.$$

The specification of the formula and the procedure of SKA are discussed in Chapter 4. So I would not repeat them here. As a matter of fact, little changes would be made to adapt the programs to other forms of ARHMM or to extend the programs to other applications. But before using these programs , make sure you read chapter 4 first.

The codes are available upon request to the author .

A.2 Implementation issues

A.2.1 Installation

Unzip ar1hmm.zip into a folder and then set a path of this folder in MATLAB, then all functions would be used in the same way as build-in MATLAB functions. The AR1HMM has been partially tested in the MATLAB 6.0 in Windows XP. No tests have been done in other versions of MATLAB and/or under other platforms, though it should work in any newer version of MATLAB and under any other OS.

A.2.2 Data structure

The main variables in AR1HMM are all MATLAB-matrix based. No specified structure has been used. They are named in the way as close to their origin as possible in Chapter 4. Here are some of them:

T The length of the bivariate observation sequence, T .

y Bivariate observation sequence, $2 \times T$, $Y_t = \begin{bmatrix} y_{t,1} \\ \\ y_{t,2} \end{bmatrix}$.

x State sequence X_t, $1 \times T$.

mu Mean vectors μ stacked as column vectors, such that mu(:,i) is the mean vector of i-th state. For two state AR1HMM, it is 2×2.

beta Autoregressive matrices $\beta^{(i)} = \begin{bmatrix} \beta_1^{(i)} & 0 \\ 0 & \beta_2^{(i)} \end{bmatrix}$. beta(:,:,i) is the autoregressive coefficient matrix of i-th state. For two state AR1HMM, it is $2 \times 2 \times 2$. Note that beta(1,2,i) and beta(2,1,i) must be zeros.

sigma covariance matrix of ϵ_t , $\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$, 2×2.

p Transition matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, where $a_{ij} = P(X_{t+1} = j | X_t = i)$, 2×2.

p0 Initial distribution matrix $\pi = \begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix}$, where $\pi_{ij} = P(X_1 = i, X_2 = j)$ are represented by p0(i,j), 2×2. Note that the sum of all entries of the p0 should be 1.

pxx Smoothed probability $P(X_t, X_{t-1}|Y)$. For two state AR1HMM, pxx is $2 \times 2 \times T$ variable and pxx(j,i,t) represents $P(X_t = j, X_{t-1} = i|Y)$. pxx are calculated by the function ar1_smooth and the procedure are described in Section (3.5).

A.2.3 Examples

A typical SKA estimation involves the following main iteration:

```
for i= 1:n_iter
```

```
        pxx=ar1_smooth(y , mu , beta , sigma, p, p0);
        [mu,beta,sigma,p,p0,x]=ar1hmm_est(pxx,y,p0);
    end
```

In each iteration, one calls the function 'ar1_smooth' to calculate the smooth probability pxx and then use pxx , along with observations y and initial probability p0, as input parameters of function 'ar1hmm_est' to re-estimate the parameter set.

The script ar1hmm_ex1 and ar1hmm_ex2 contains some codes to demonstrate the usage of AR1HMM. ar1hmm_ex1 contains the most essential codes of parameter initialization, re-estimation procedure and estimation storage. ar1hmm_ex2 provides a more elaborate example of generating bivariate ARHMM observations , automatic initialization, etc.

A.2.4 Initialization

Initialization of original parameters plays an important role in AR1HMM because of the local minimum problems. init_ar1hmm provides a simple and very basic way of initialization. The observations are classified into two states based on the values of their norms. The threshold can be set in the init_ar1hmm.m . The recommendation here is to visualize the data first and then determine the threshold through observation and judgement.

One frequent question would be : Does the init_ar1hmm works well? Regretfully

the answer is 'no' in most cases. One must realize that the initialization is heavily depends on the application considered. For a better initialization, one may consider to use the cluster analysis in MATLAB toolbox.

A.3 Alphabetical list of functions

ar1_loglike Calculate the log-likelihood of AR(1)HMM based on the forward procedure. Because the forward variable 'sai' are normalized in each step to avoid the overflow problem, 'logscale' is the real log-likelihood value instead of 'Loglv'.

ar1_smooth Calculate the smoothed probabilities $P(X_t, X_{t-1}|Y, \lambda)$ based on the procedure described in Section 3.5.

ar1hmm_chk Verify the parameters of an ar1HMM and returns the number of states.

ar1hmm_est Estimate the parameter set of AR(1)HMM based on the SKA. The implementation of Section 4.2.

ar1hmm_ex1 The first example of AR1HMM.

ar1hmm_ex2 The second example of AR1HMM

ar1hmm_gen Generate bivariate two-state AR(1)HMM observations based on the parameter set in 'init_par.mat'. If 'init_par.mat' doesn't exist, then use the parameter values in the code.

init_ar1hmm Automatic initialize the AR(1)HMM parameters.

scaler Sub-function used in 'ar1_loglike' to rescale the forward variable to avoid the overflow.

References

1. Bowerman and O'Connell, *Forecasting And Time Series - An Applied Approach*(Third Edition). Duxbury : 1993.

2. G.E.P. Box and G.M.Jenkins, *Time Series Analysis - forcasting and control*. Holden-Day Inc : 1976.

3. C.Chatfield, *The Analysis of Time Series-An Introduction* (Fourth Edition). Chapman and Hall: 1989.

4. Olivier Cappe , *H2M : A set of MATLAB/OCTAVE functions for the EM estimation of mixtures and hidden Markov models*. CNRS-URA 820 : Aug. 2001 .

5. Dempster, A.P., Laird, N.M., and Rubin D.B. , *Maximum Likelihood Estimation for Incomplete Data via the EM algorithm* J.R. Statist. Soc. B Vol 39, p.p.1-38.: 1977

6. Rakesh Dugad, U.B.Desai *A tutorial On Hidden Markov Model* Technical Report SPANN-96.1: 1996

7. E.A.Elgmati *Hidden Markov Models with Applications to Autoregressive Time Series Models*. Master Thesis , Dept. of Math and Stat , Univ. of Saskatchewan : 1999.

8. Geoffery Grimmett, David Stirzaker, *Probability and Random Process*. Clarendon Express , Oxford : 1982.

9. James D.Hamilton , *A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle*. Econometrica , Vol. 57 No.2 p.p.357-384 : Mar , 1989 .

10. James D.Hamilton , *Analysis of Time Series subject to Change in Regime*. J. Econometrics 45, p.p 39-70. : 1990 .

11. James D.Hamilton , Gabriel Perez-Quiros, *What Do the Leading Indicators Lead?* J.Business , 1996 , vol.69 , no.1, p.p.27-49. : 1996.

12. James D.Hamilton and Gang Lin, *Stock Market Volatility and the Business Cycle*. J. of Applied Econometrics , Vol. 11 , p.p.573-593 : 1996 .

13. James D.Hamilton , Baldev Raj (Editors), *Advances in Markov-Switching Models*. Heidelberg , New York. Physica-Verl :2002.

14. B.H.Juang *On the Hidden Markov Model and Dynamic Time Warping Speech Recognition - A Unified View*. AT&T Tech. J. Vol.63 No.7 : 1984.

15. B.H.Juang and Lawrence R. Rabiner *A Probabilistic Distance Measure for Hidden Markov Models*. AT&T Tech. J. Vol.64 No.2 , p.p. 391-408 : 1985.

16. B.H.Juang and Lawrence R. Rabiner *Mixture Autoregressive Hidden Markov Models for Speech Signals*. IEEE Trans. Vol. ASSP-33, No.6 , p.p. 1404-1413: Dec. 1985 .

17. B.H.Juang , Stephene Levinson and M.M.Sonphi *Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains*. IEEE Tran. Vol. IT-32 ,No.2 p.p. 291–298 : March 1986.

18. B.H.Juang and L.R. Rabiner, *The segmental k-means algorithm for estimating the parameters of hidden Markov models* IEEE Trans. Accoust., Speech, Signal Processing, vol. ASSP-38, no.9, pp.1639-1641 : 1990 .

19. Brian G.Leroux, *Maximum-Penalized-Likelihood Estimation for Independent and Markov Dependent Mixture Models* Biometerics 48, p.p. 545-558 : 1992 .

20. N.M. Laird, *Nonparametric Maximum Likelihood Estimation of a Mixing Distribution* J. Amer. Stat. Asso. 73, p.p. 805-811: 1978 .

21. L.R.Liporace, *Maximum likelihood estimation for multivariate observations of Markov Sources* IEEE Trans. Info. Theory, 28, p.p. 729 : 1982.

22. R.B.Lyngso and C.N.S.Pedersen and H.Nielsen, *Metrics and similarity measures for hidden Markov models.* AAAI : 1999.

23. Douglas C.Montgomery, Lynwood A.Jonhnson, Jone S.Gardiner, *Forecasting & Time Series Analysis* (Second Edition). McGrew-Hill,Inc: 1990.

24. The MathWorks.Inc, *Using Matlab (Version 6)* . 2000.

25. I.L.MacDonald and W.Zucchini, *Hidden Markov and Other Models for Discrete-valued Time Series.* Chapman&Hall : Jan, 1997 .

26. Z.Psaradakis N.Spagnolo, *On The Determination Of The Number Of Regimes In Markov-Switching Autoregressive Models.* Journal of Time Series Analysis : 2004.

27. W. Qian and D.M. Titterington *Estimation of parameters in hidden Makov models.* Phil. Trans. R. Soc. Lond. A 337, p.p. 407-428: 1991.

28. Lawrence R. Rabiner *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proc. IEEE, Vol. 77 , No.2 , p.p.257-286 : Feb, 1989.

29. Robertson, S. Kirshner, P. Smyth. *Hidden Markov models for modeling daily rainfall occurrence over Brazil*, Technical Report UCI-ICS 03-27 p.p.29-56 : November 2003.

30. T.R. Turner , M.A. Camron And P.J.Thomson *Hidden markov chains in generalized linear models.* Canada.J. Stat. Vol. 26, No.1, p.p. 107 C 125 :1998.

31. William W.S.Wei, *Time Series Analysis - Univariate and Multivariate Methods*. Addison-Wesley: 1990.