A REVIEW AND APPLICATION OF HIDDEN MARKOV MODELS AND DOUBLE CHAIN MARKOV MODELS

Michael Ryan Hoff

A Dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science

Johannesburg, 2016

DECLARATION

I declare that this Dissertation is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

_____

(Signature of candidate)

_____day of_____ 20_____in_____

ABSTRACT

Hidden Markov models (HMMs) and double chain Markov models (DCMMs) are classical Markov model extensions used in a range of applications in the literature. This dissertation provides a comprehensive review of these models with focus on i) providing detailed mathematical derivations of key results - some of which, at the time of writing, were not found elsewhere in the literature, ii) discussing estimation techniques for unknown model parameters and the hidden state sequence, and iii) discussing considerations which practitioners of these models would typically take into account.

Simulation studies are performed to measure statistical properties of estimated model parameters and the estimated hidden state path - derived using the Baum-Welch algorithm (BWA) and the Viterbi Algorithm (VA) respectively. The effectiveness of the BWA and the VA is also compared between the HMM and DCMM.

Selected HMM and DCMM applications are reviewed and assessed in light of the conclusions drawn from the simulation study. Attention is given to application in the field of Credit Risk.

Contents

List of Figures

List of Tables

Notation and Abbreviations

Although notation and abbreviations are defined as they are introduced, the below list of commonly used notation and abbreviations may be helpful.

Notation

The notation listed assumes discrete-time, discrete-state space and discrete-signal space models which are time homogeneous.

S_k	Output (signal) observed at time k
$\mathbf{S}_n$	The sequence of observed outputs (signals) for the first n time points. This will be $(S_1, S_2, \ldots, S_n)$ for a HMM and $(S_0, S_1, S_2, \ldots, S_n)$ for a DCMM.
X_k	State occupied by a Markov chain at time k
$\mathbf{X}_n$	The sequence of states visited by a process for the first n time points. This will be $(X_1, X_2, \ldots, X_n)$ for both the HMM and the DCMM.
S	The state space, that is the set of all possible states which can be visited by the process
δ	The signal space, that is the set of all possible signals which can be observed
p_i	The unconditional initial state probabilities, that is $p_i = P(X_1 = i)$ for $i \in S$
$p_i(k)$	The unconditional state probabilities at time k, that is $p_i(k) = P(X_k = i)$ for $i \in S$
π_i	The limiting steady state probability for state $i \in S$
π	Vector containing the limiting steady state probabilities, that is $\pi = \{\pi_i\}$
p_{ij}	The probability of a one-step transition from state $i \in S$ to state $j \in S$. For a given time point k, $p_{ij} = P(X_{k+1} = j \mid X_k = i)$.
b_{jm}	The probability of observing output (signal) $\nu_m \in \delta$, conditional on a HMM being in state $j \in S$. For a given time point k, $b_{jm} = P(S_k = \nu_m \mid X_k = j)$.
$b_{lm}^{(j)}$	The probability of observing output (signal) $\nu_m \in \delta$, conditional on a DCMM being in state $j \in S$ and the previous observed output (signal) being $\nu_l \in \delta$. For a given time point k, $b_{lm}^{(j)} = P(S_k = \nu_m \mid X_k = j, S_{k-1} = \nu_l)$.
$\mathbf{a}$	Vector containing the unconditional initial state probabilities, that is $\mathbf{a} = \{p_i\}$

$\mathbf{P}$	Matrix containing the one-step state transition probabilities, that is $\mathbf{P} = \{p_{ij}\}$
$\mathbf{B}$	Matrix containing the output (signal) probabilities for a HMM, that is $\mathbf{B} = \{b_{ij}\}$
$\mathbf{B}^{(j)}$	Matrix containing the output (signal) transition probabilities for a DCMM, conditional on state $j \in S$. That is $\mathbf{B}^{(j)} = \{b_{lm}^{(j)}\}$.
λ	The complete parameter set for a HMM or a DCMM. That is $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$ where, for the DCMM, $\mathbf{B}$ represents a stacking of the $\mathbf{B}^{(j)}$ matrices.
$F_k(j)$	The forward equation, defined as $P(\mathbf{S}_k = \mathbf{s}_k, X_k = j\|\lambda)$
$B_k(i)$	The backward equation, defined as $P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n\|X_k = i, \lambda)$
$V_k(j)$	The Viterbi equation, defined as $\max_{i_1,\ldots,i_{k-1}} P(\mathbf{X}_{k-1} = (i_1, \ldots, i_{k-1}), X_k = j, \mathbf{S}_k = \mathbf{s}_k\|\lambda)$

Abbreviations

AIC	Akaike information criterion
BIC	Bayesian information criterion
BWA	Baum-Welch Algorithm
DCMM	Double Chain Markov model
EM algorithm	Expectation Maximization algorithm
HMM	Hidden Markov Model
MTD model	Mixture transition distribution model
VA	Viterbi Algorithm

Chapter 1

Introduction

1.1 An Overview

Markov models are a family of stochastic processes familiar to many statisticians. The underlying assumption of Markov models is that the state process possesses the Markov property (defined in Section 1.4). Importantly, the state process of Markov models is directly observable.

A hidden Markov model (HMM) is an extension of the ordinary Markov model since the evolution of the state process is governed by the Markov property. Now, however, the states visited are no longer known or observable. Instead only signals (or outputs), which are produced by the states which have been visited, are observable. It is assumed that these signals are emitted by the state process as follows: each state in the state space has a probability distribution defined on the set of all possible signals, and at each time point the current state will emit a signal according to this distribution.

While the sequence of signals observed is dependent on the sequence of states visited, there is no direct dependence structure between successive signals emitted - that is, conditional independence exists between the signals.

However, in some instances an assumption of conditional independence of the signals might not be justified. In such cases it would be advantageous to assume a process which satisfies the assumptions of a HMM, but where, for a given time point, the signal emitted is not only dependant on the current state, but also on the previous signal(s) which has been observed. An example of such an extension of the HMM is the double chain Markov model (DCMM). In particular, the DCMM assumes that the signal process also possesses the Markov property. That is, both the state and signal processes are driven by the Markov property, where the signal process is also dependent on the states which the state process visits.

The fundamentals of the Markov model, the HMM and the DCMM are summarised in three plots presented in [10] and are reproduced in the next page in Figures 1.1 - 1.3.

Figure 1.1 shows a Markov model whereby the output (the state process) possesses the Markov property and is directly observable. Figure 1.2 shows a HMM whereby the output process is dependent on the state process (which possesses the Markov property, but is not observable) but is conditionally independent of the previous outputs. Figure 1.3 shows a DCMM whereby the output process is not only dependent on the state process (which possesses the Markov property, but is not observable) but is also conditionally dependent on the previous outputted value through the Markov property.

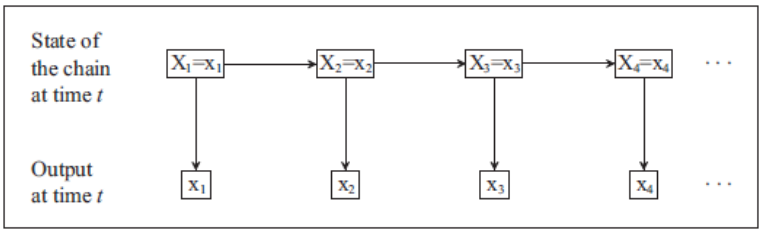


Figure 1.1: Representation of a Markov chain.

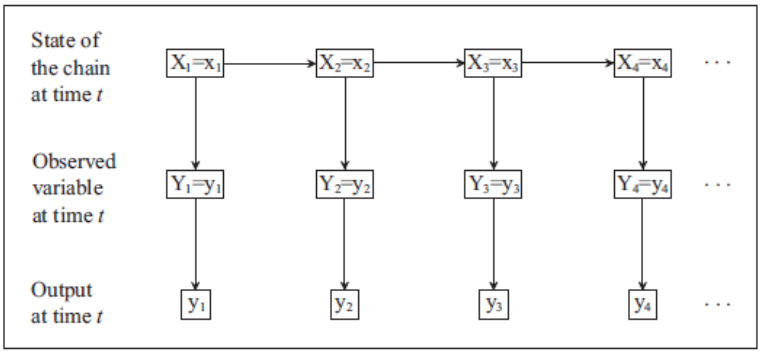


Figure 1.2: Representation of a hidden Markov model

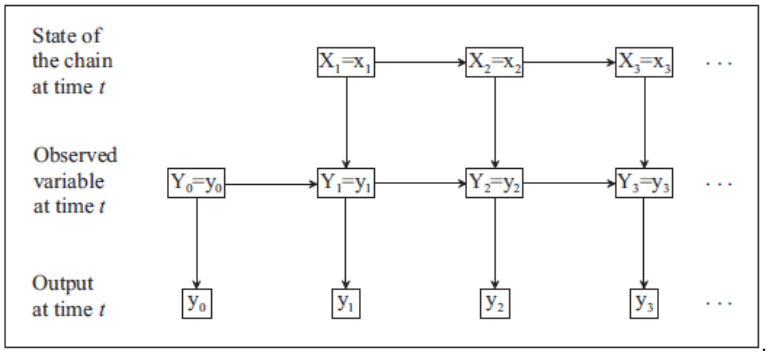


Figure 1.3: Representation of a double chain Markov model

1.2 Objectives of the Dissertation

The primary objectives of this dissertation are to

- Provide a detailed theoretical review of HMMs including a review of estimation techniques which may be used to estimate model parameters and the underlying hidden state sequence. While alternative HMM specifications will be overviewed, the focus of this dissertation will be on discrete-time, discrete-state space and discrete-signal space HMMs. Attention will also be given to the mathematical derivation of key results.

- Explore and detail how the mathematical framework of the HMM can be extended to formulate the mathematical framework for the DCMM. As with the HMM, the focus of the research will be on discrete-time, discrete-state space and discrete-signal space DCMM.

- Perform a comprehensive simulation exercise which explores the behaviour of model parameter estimation and the estimation of the underlying hidden state sequence for both the HMM and DCMM.

- Provide a review of selected HMM and DCMM applications which are documented in the literature, and assess several of these applications in light of conclusions drawn from the simulation study mentioned above. Attention is given to the application of HMMs and DCMMs in the field of Credit Risk.

This dissertation is structured as follows. The remainder of Chapter 1 provides a brief overview of stochastic processes before introducing Markov models - in particular the discrete-time, discrete-state space Markov model (also known as the Markov chain). This is followed by a detailed discussion of the discrete-time, discrete-state space and discrete-signal space HMM in Chapters 2 to 5. Alternative HMM specifications are

also overviewed in these chapters. Chapter 6 then proceeds to detail the discrete-time, discrete-state space and discrete-signal space DCMM. Simulation exercises for HMMs and DCMMs, and conclusions which can be drawn from these simulations, are discussed in Chapter 7. Chapter 8 provides a review of selected documented applications of HMMs and DCMMs. Concluding remarks are then given in Chapter 9. Finally the appendices of this dissertation provide mathematical detail of key results within the HMM and DCMM framework.

1.3 Stochastic Processes

The material outlined in this section is covered in many introductory references within the literature, see for example Sections 2.1 and 2.8 of [41].

Markov models, HMMs and DCMMs all fall under the broad field of stochastic processes. In order to adequately describe these models, an overview of the concept of a stochastic process is therefore required.

To begin, a random variable is defined as a variable whose value results from the measurement of some type of random process. It is typically a function which will associate a unique numerical value with every possible outcome of the random process. The value of the random variable is not known in advance and will vary with each realisation of the random process. However, the probability distribution of the random variable is known (or may be inferred) and can therefore be used to describe probabilities of interest regarding the random variable.

For example, consider the random process of tossing a coin. The possible outcomes for this process are $\Omega = \{\text{head, tail}\}$. The random variable Y may be introduced for this random process as follows

$$Y(\omega) = \begin{cases} 0, & \text{if } \omega = \text{heads;} \\ 1, & \text{if } \omega = \text{tails.} \end{cases}$$

Before each coin toss, although it is known that $Y \in \{0, 1\}$, the exact value that will be realised for the random variable Y is unknown. However, if the coin is unbiased, the probability mass function for Y is known and given by

$$P(Y = y) = \begin{cases} \frac{1}{2}, & y = 0; \\ \frac{1}{2}, & y = 1. \end{cases}$$

Finally, random variables are either classified as discrete (a random variable that may assume values from a countable set) or as continuous (a variable that may assume any numerical value in an interval or collection of intervals on the real line).

A stochastic process $\{X(t) : t \in T\}$ is then a sequence of random variables, indexed by the set T, that describes the evolution of some physical process over the set T. That is, for each $t \in T$, $X(t)$ is a random variable. Typically for a stochastic process, some form of dependence structure exists among the random variables $X(t)$, $t \in T$. These dependence relationships dictate the manner in which the random variables will evolve over t and thus play a role in characterising the stochastic process.

In most applications, the index t is interpreted as time. $X(t)$ is then referred to as the state of the process at time t. For example, $X(t)$ could represent:

- the number of customers in a supermarket at time t,
- the total number of customers that have entered a supermarket up to time t,
- the total sales amount registered at a supermarket up to time t,
- the amount of time that a customer queues at the cashier of a supermarket at time t,
- the size of a bacteria colony after elapsed time t,
- the number of calls arriving at a telephone exchange during a time interval $[0, t)$,
- the return yielded by an asset at time t, etc.

The set T is called the index set of the process. If T is a countable set the stochastic process is said to be a discrete-time process. If T is an interval of the real line (uncountable) the stochastic process is said to be a continuous-time process. For example:

- $\{X_n : n = 0, 1, 2, \ldots\}$ is a discrete-time stochastic process indexed by the non-negative integers,
- $\{X(t) : t \geq 0\}$ is a continuous-time stochastic process indexed by the non-negative real numbers.

The state space S of a stochastic process is the set of all possible values which the random variables $X(t),\ t \in T$, can assume. That is $X(t) \in S$ for each $t \in T$. If S is a countable set the stochastic process is said to be a discrete-state process. If S is an interval of the real line (uncountable) the stochastic process is said to be a continuous-state process. For example:

- if $S = \mathbb{Z}$ then $\{X(t) : t \in T\}$ is a discrete-state stochastic process,
- if $S = \mathbb{R}$ then $\{X(t) : t \in T\}$ is a continuous-state stochastic process.

Based on the above, four distinct categories of stochastic processes are possible. These are shown, together with examples, in Table 1.1.

In summary, a stochastic process is a collection of random variables which, through dependence relationships among the random variables, describe the evolution of some physical process over some index set (typically time). As such, a stochastic process is therefore characterised by:

- the nature of the index set T,
- the nature of the state space S,
- the dependence relationships amongst the random variables $X(t),\ t \in T$.

Process	Examples
Discrete-time, Discrete-state	Simple random walk Gambler's Ruin Markov chain
Discrete-time, Continuous-state	Time series process Markov process
Continuous-time, Discrete-state	Generalised random walk Poisson process Yule process Birth-and-death process Continuous-time Markov chain
Continuous-time, Continuous-state	Brownian motion/Wiener process Diffusion process

Table 1.1: Types of stochastic processes.

1.4 Discrete Time Markov Chain

The material outlined in this section is covered in many introductory references within the literature, see for example Sections 4.1 to 4.4 of [41].

It has been previously mentioned that the HMM and the DCMM are extensions of the family of stochastic processes known as Markov models. In order to completely define the HMM and DCMM, a basic understanding of Markov models is therefore first required. As outlined in the previous section (in Table 1.1), different types of Markov models (based on the nature of the state space and the index set) are used in applications. However, the distinguishing property of any Markov model (which differentiates it from other types of stochastic process models) is the fact that the dependence relationship among the states which are visited is driven by the Markov property. The Markov property states that the probability that the process will enter a given state at a given future time is independent of past states which have been visited and depends only on the state that the process is currently in. The Markov property will thus drive the evolution of the states in Markov models (for a

mathematical representation, see equation (1.1) later).

This dissertation will be focusing on the discrete-time, discrete-state space HMMs and DCMMs, so only the discrete-time, discrete-state space Markov model (otherwise known as the Markov chain) will be further detailed in this section. Also, since the focus of this dissertation is on the HMM and DCMM, the primary focus of this section will be to simply provide widely-used results for the Markov chain. Many of the numerous sources which discuss Markov chains in the literature contain proofs for these results, for example [41].

A Markov chain is a discrete-time stochastic process $\{X_n : n = 1, 2, \ldots\}$ with X_n, the state at a given time point n, taking on a countable number of possible values. That is the state space is discrete and will for the remainder of this dissertation be denoted by S, where

$$S = \{1, 2, \ldots, M\}, \text{ if the state space is finite, or}$$
$$S = \{1, 2, \ldots\}, \text{ if the state space is infinite (but countable),}$$

where 1 represents state 1; 2 represents state 2; and so on. Thus $X_n = i, \text{ where } i \in S$, implies that the Markov chain is in state i at time n.

It should be noted that for the purposes of this dissertation, the initial state of the process is denoted by X_1 (since this is the notation which is widely used in the literature for the HMM).

Furthermore, let the transition probabilities and the initial state probabilities be respectively denoted by

$$p_{ij}(m, n) = P[X_n = j | X_m = i], \text{ and}$$
$$p_i = P[X_1 = i]\,.$$

As was previously mentioned, the distinguishing property of a Markov chain is that the state transitions are governed by the Markov property. This is expressed mathe-

matically, for the Markov chain, as follows

$$P[X_{m+l} = j | X_1 = i_1, X_2 = i_2, \ldots, X_m = i] = P[X_{m+l} = j | X_m = i] \quad (1.1)$$

for states $i_1, \ldots, i_{m-1}, i, j \in S,$ and $l \in \{1, 2, \ldots\}$.

If $p_{ij}(m, m+1) = p_{ij}(1)$ for all $m \in \{1, 2, \ldots\}$, then the Markov chain is said to be time homogeneous. This dissertation will assume time homogeneity throughout. It then holds that, for a given $l \in \{1, 2, \ldots\}$, the l-step transition probability satisfies

$$P[X_{m+l} = j | X_m = i] = p_{ij}(l), \quad \text{for all } m \in \{1, 2, \ldots\}.$$

That is, under the assumption of time homogeneity, the transition probabilities do not depend on time points m or $m + l$, but only on the time interval l.

Since p_i and $p_{ij}(l)$ are probabilities, the following constraints must hold:

$$\begin{aligned} &\sum_{i \in S} p_i = 1 \\ &p_i \geqslant 0, \qquad \text{for } i \in S, \text{ and} \end{aligned} \quad (1.2)$$

$$\begin{aligned} &\sum_{j \in S} p_{ij}(l) = 1, \qquad \text{for } i \in S \text{ and } l \in \{1, 2, \ldots\} \\ &p_{ij}(l) \geqslant 0, \qquad \text{for } i, j \in S \text{ and } l \in \{1, 2, \ldots\}. \end{aligned} \quad (1.3)$$

The one-step transition probability is defined as the probability of going directly from state i to state j in one transition and is obtained by setting $l = 1$, that is

$$p_{ij} = p_{ij}(1) = P[X_{m+1} = j | X_m = i], \quad \text{for all } m \in \{1, 2, \ldots\}.$$

Define matrix $\mathbf{P}$ to be the matrix containing the one-step transition probabilities. That is $\mathbf{P} = \{p_{ij}\}$.

A given transition probability $p_{ij}(n + m)$ can be expressed (see for example [41]) in the following way:

$$p_{ij}(m + n) = \sum_{k \in S} p_{ik}(m) p_{kj}(n), \quad \text{for all } n, m \geqslant 1 \text{ and } i, j \in S. \quad (1.4)$$

The equations collected in (1.4) are referred to as the Chapman-Kolmogorov equations. For a given n, let $\mathbf{P}^{(n)}$ denote the matrix containing the n-step transition probabilities. That is $\mathbf{P}^{(n)} = \{p_{ij}(n)\}$. Then equation (1.4) implies that

$$\mathbf{P}^{(m+n)} = \mathbf{P}^{(m)} \,.\, \mathbf{P}^{(n)}, \tag{1.5}$$

where the dot represents matrix multiplication.

From equation (1.5), for any $l \in \{1, 2, \ldots\}$,

$$\mathbf{P}^{(l)} = \mathbf{P}^{l}. \tag{1.6}$$

This can be proven by setting $m = n = 1$ in equation (1.5) and then making use of mathematical induction. Thus, the matrix containing the l-step transition probabilities can be obtained by multiplying the matrix $\mathbf{P}$ by itself l times; and so a transition probability of any arbitrary step length can be derived from the one-step transition probabilities. As such, the one-step transition probabilities is a crucial aspect of the Markov chain. The way in which these transition probabilities are chosen will vary depending on the application.

The unconditional probabilities for the Markov chain are defined as follows:

$$p_i(n) = P(X_n = i), \qquad \text{for } i \in S \text{ and } n \in \{1, 2, \ldots\}\,.$$

The vector of unconditional probabilities at time n is denoted by

$$\mathbf{p}(n) = (p_1(n), p_2(n), \ldots, p_i(n), \ldots)\,.$$

Using the well known statistical results of the partition rule for probabilities and the multiplication rule for probabilities (these are defined later in this dissertation in equations (2.9) and (2.10) respectively), it follows that

$$\mathbf{p}(m+n) = \mathbf{p}(m)\,.\,\mathbf{P}^{(n)}\,. \tag{1.7}$$

Equation (1.7) implies

$$\mathbf{p}(n) = \mathbf{p}(1) \,.\, \mathbf{P}^{(n-1)} = \mathbf{p}(1) \,.\, \mathbf{P}^{n-1}, \tag{1.8}$$

where $\mathbf{p}(1)$ is the vector containing the initial unconditional probabilities of the process.[1] This result shows that the evolution of the Markov chain is determined completely by the distribution of the initial probabilities $\mathbf{p}(1)$ and the one-step transition matrix $\mathbf{P}$.

In certain applications of the Markov chain it may well be of interest to determine the value of $\mathbf{p}(n)$ as n becomes large (that is as $n \to \infty$). To this end, vector π is said to contain the limiting probabilities for the Markov chain (also referred to as the stationary distribution) if for each $j \in S$ it satisfies

$$\begin{aligned} & \pi_j \geq 0 \quad \text{and} \\ & \sum_{j \in S} \pi_j = 1 \quad \text{and} \\ & \pi_j = \sum_{i \in S} \pi_i p_{ij} \\ \Rightarrow \quad & \pi = \pi \,.\, \mathbf{P} \,. \end{aligned} \tag{1.9}$$

Now, if $\lim_{n \to \infty} \mathbf{p}(n)$ exists then $\lim_{n \to \infty} \mathbf{p}(n) = \pi$. This follows from $\mathbf{p}(n+1) = \mathbf{p}(n) \,.\, \mathbf{P}$ (which can easily be verified using equation (1.7)).

A general result suggested by [46], which can conveniently be used to determine the limiting probabilities (when they exist) is the following:

$$\pi = \mathbf{1}(\mathbf{I}_m - \mathbf{P} + \mathbf{U}_m)^{-1}, \tag{1.10}$$

[1] For consistency with other models described in this dissertation, the state process is assumed to begin at time 1. Hence the initial state is defined at time 1.

where m is the number of states in the state space,

$\mathbf{P}$ is the transition probability matrix,

$\mathbf{1}$ is a m-dimensional row vector of ones,

$\mathbf{I}_m$ is the $m \times m$ identity matrix,

$\mathbf{U}_m$ is the $m \times m$ matrix of ones.

This can easily be proven as follows:

$$\begin{aligned} \pi &= \pi \,.\, \mathbf{P} \\ \pi \,.\, \mathbf{I}_m - \pi \,.\, \mathbf{P} + \pi \,.\, \mathbf{U}_m &= \pi \,.\, \mathbf{U}_m \\ \pi \,(\mathbf{I}_m - \mathbf{P} + \mathbf{U}_m) &= \mathbf{1} \\ \pi &= \mathbf{1}(\mathbf{I}_m - \mathbf{P} + \mathbf{U}_m)^{-1}. \end{aligned}$$

It should be noted that in general the limiting probabilities of the Markov chain are not guaranteed to exist, and if they do exist, they need not be unique. Conditions defining when these limiting probabilities will exist and when they can be guaranteed to be unique are discussed in [41]. The situation is somewhat simplified if the state space of the Markov chain is finite. In fact, if S is finite, and the Markov chain is aperiodic (if returns to a given state $i \in S$ can occur at non-regular time intervals) and irreducible (a Markov chain is said to be irreducible if there exists a positive integer n_{ij} such that $p_{ij}(n_{ij}) > 0$ for each possible (i, j) pair, where $i, j \in S$), then the limiting probabilities for the Markov chain will exist and will be unique.

The following is also concluded in numerous sources in the literature (see for example [41]) for the limiting probabilities of a Markov chain :

- π_j represents the long-run proportion of time that the process will be in state j.

- π_j is independent of the current state that the process is in.

- If p_j is chosen to be π_j for each $j \in S$, then $p_j(n) = \pi_j$ for each $n \in \{2, 3, \ldots\}$. For this reason the limiting probabilities are often also referred to as the stationary probabilities.

- Finally if for state $j \in S$, m_{jj} is defined to be the expected number of transitions until a Markov chain in state j will return to state j, then $\pi_j = \frac{1}{m_{jj}}$. Alternatively, $m_{jj} = \frac{1}{\pi_j}$.

Numerous extensions of Markov models exist in the literature. The next chapter will explore one such extension, namely the hidden Markov model.

Chapter 2

An Introduction to the Hidden Markov Model

2.1 Defining the Hidden Markov Model

A hidden Markov model (HMM) is a doubly embedded stochastic process whereby the underlying stochastic process (that of the states) possesses the Markov property. This stochastic process of the states, denoted by $\{X(t) : t \in T\}$, is at no point known or observable. However, a second stochastic process $\{S(t) : t \in T\}$, that being the process of observations (or signals), is observable and is driven by the unobservable state process in the following way: it is assumed that at each time point, the current state emits a signal according to a probability distribution defined on the set of all possible signals for that state. In this way, the first stochastic process (that of the states) drives the second stochastic process (that of the signals). However, while the sequence of signals is dependent on the sequence of states visited, it is important to note that there is no direct dependence between successive signals emitted - that is, given the current state, each new signal emitted is conditionally independent of all previous signals emitted.

Due to the relationship which exists between the state and signal processes, a signal

sequence which has been observed can be used to infer the most likely sequence of states already visited by the HMM and estimate, among other probabilities of interest, the probability that certain states will be visited in the future.

Applications of the HMM are numerous and examples of applications in the literature extend to a variety of fields of study. A selection of these applications are discussed further in Chapter 8 of this dissertation.

Owing to the fact that the state process of the HMM is governed by the Markov property, the HMM can be seen as an extension of the Markov model (discussed in Section 1.4). While the name 'hidden Markov model' occurs commonly in the literature it is by no means the only name used. Other terms which also appear in the literature for such models include 'hidden Markov process', 'Markov-dependent mixture', 'Markov-switching model', 'models subject to Markov regime' and 'Markov mixture model'.

2.1.1 A Simple Example of a Hidden Markov Model

An illustrative example of the HMM is given below. This is based on an example given in [41] - see page 257.

Consider a machine that produces a single product at each time point. At each time point the machine can be in one of two conditions, either in a good condition (state 1), or in a poor condition (state 2) - where it is assumed that the process of the condition of the machine over time possesses the Markov property. If the machine is in state 1 during the current period, it will (independent of the previous states) remain in state 1 during the next period with probability 0.9. State 2 is an absorbing state (if the machine is in poor condition it will, independent of the previous states, remain in poor condition for all future periods). Now suppose it is impossible to observe what condition the machine is in directly, but it is possible to observe the quality of the product that the machine produces (the product can either be satisfactory

or defective). Furthermore, suppose that the machine will produce a product of satisfactory quality when it is in state 1 with probability 0.99, while it will produce a satisfactory product with probability 0.96 if it is in state 2.

The above is then a simple example of a HMM. This follows since the process of the condition of the machine (the sequence of the states) has the Markov property and is unobservable, while the process of the products produced (or signals) is observable and driven entirely by the sequence of states, through a probabilistic distribution defined on the set of possible signals for each state.

2.1.2 Elements of a Discrete-Time Hidden Markov Model

HMMs are well described in the literature, see for example [37], [41] and [46] - which have been used as the basis for the material presented in this section and Section 2.1.3. In particular, this section will define and detail the various components of the discrete-time HMM. To this end, assume that $\{X_n : n = 1, 2, \ldots\}$ represents the unobservable state process and that $\{S_n : n = 1, 2, \ldots\}$ represents the observable signal process. Since the process of the states $\{X_n : n = 1, 2, \ldots\}$ represents a Markov chain, the results and findings of Section 1.4 will apply to $\{X_n\}$. The various components of the discrete-time HMM are now defined below.

State space S: As was defined for Markov chains (see Section 1.4), let S represent the state space. It will be assumed for the remainder of this dissertation that the state space for the HMM is discrete, That is,

$$S = \{1, 2, \ldots\},$$

where 1 represents state 1; 2 represents state 2; and so on.

In example 2.1.1, $S = \{1, 2\}$, where 1 = good condition (state 1), and
2 = poor condition (state 2).

Initial state probabilities p_i: As was defined for Markov chains let $p_i = P[X_1 = i]$, for all $i \in S$, represent the unconditional initial state probabilities, subject to the constraints given in equation (1.2). Further define $\mathbf{a}$ to be the vector containing all these initial probabilities.

One-step state transition probabilities p_{ij}: It is assumed for the purposes of this dissertation that state transitions are time homogeneous.

Therefore, as was the case for Markov chains, $p_{ij} = P[X_{m+1} = j | X_m = i]$ (for all $m = 1, 2, \dots$), and $\mathbf{P} = \{p_{ij}\}$ is the one-step transition probability matrix containing all these transition probabilities for the state sequence. These one-step transition probabilities are subject to the constraints given in equation (1.3).

In example 2.1.1, $\mathbf{P} = \begin{pmatrix} 0.9 & 0.1 \\ 0 & 1 \end{pmatrix}$.

Signal probabilities: The signals observed can either be from a discrete or a continuous space (or a mixture of both).

Discrete signal space: Let $\delta_n = \{v_1, v_2, \dots\}$ be the set of all possible signals which can be emitted at time n, for $n = 1, 2, \dots$, and let S_n be the signal emitted at time n, where $S_n \in \delta_n$. Further define

$$b_{ik}(n) = P[S_n = v_k | X_n = i]$$

to be the probability of observing signal v_k (where $v_k \in \delta_n$) at time n, given that the process is in state i at time n.

For a discrete signal space, if time homogeneity is assumed then the set of all possible signals which can be observed will not change over time. In such cases, let δ be the signal space for all $n = 1, 2, \dots$. Then,

$$\delta = \delta_n, \text{ and } S_n \in \delta, \text{ for } n = 1, 2, \dots .$$

Under these time homogeneous conditions for the signals, define

$$b_{ik} = P[S_n = v_k | X_n = i]\,,$$

which does not depend on the time n.

Also define the signal probability matrix to be $\mathbf{B}$, where the i^{th} row and k^{th} column of $\mathbf{B}$ is b_{ik}. That is $\mathbf{B} = \{b_{ik}\}$.

In example 2.1.1, $\delta \in \{v_1, v_2\}$ for $n = 1, 2, \ldots$, where v_1 = satisfactory, and v_2 = defective,

and $\mathbf{B} = \begin{pmatrix} 0.99 & 0.01 \\ 0.96 & 0.04 \end{pmatrix}$.

Continuous signal space: When the signal space, δ_n, is a continuous space (which is not necessarily time homogeneous),

$$b_i(S_n) = f(S_n | X_n = i)$$

is the probability density function (pdf), defined over δ_n, for the emitted signal at time n, given that the process is in state i at time n.

Unless stated otherwise, the remainder of this dissertation will make reference to a discrete-time, discrete-state and discrete-signal HMM where both the state and signal processes are assumed to be time homogeneous. From the above, a HMM is fully described by $\mathbf{P}$, $\mathbf{B}$, and $\mathbf{a}$. For convenience, the compact notation

$$\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$$

will be used to denote the complete parameter set of a HMM.

For the purpose of readability, $S_k = s_k$ will at times be written in the shortened form S_k. Similarly X_k will at times be used to represent $X_k = i_k$. In addition the terms 'signal sequence' and 'observation sequence' will be used interchangeably in

this dissertation without implying different meanings.

As a final observation, it should be noted that the time homogeneous, discrete-time and discrete-state Markov chain is a special case of the time homogeneous, discrete-time, discrete-state and discrete-signal HMM. This is proven in Appendix A of this dissertation.

2.1.3 Further Properties Regarding the Hidden Markov Model

As has been mentioned, it is assumed that the state process of a HMM possesses the Markov property. Furthermore, since the state process drives the signal process (and not vice a versa), no signal which has already been emitted will play a role in what a future state will be. Therefore, the following holds for $t = 1, 2, 3, \ldots$

$$P(X_{n+t} = j | S_1, X_1, \ldots, S_n, X_n = i) = P(X_{n+t} = j | X_n = i). \tag{2.1}$$

It should however be noted that the following does *not* hold:

$$P(X_{n+t} = j | S_1, \ldots, S_n) = P(X_{n+t} = j), \quad \text{where } t = 1, 2, 3, \ldots . \tag{2.2}$$

The reason why equation (2.2) does not hold is that while the signal process does not drive the state process in any way, the signal sequence which has been observed does hold information as to what the current state is - which (due to the Markov property) will influence the probability of the state at time $n + t$. Equation (2.2) will in fact be correctly defined later in Section 3.4.3.

Another key assumption for the HMM is that the probability of a signal being emitted is dependent only on the state of the HMM at the time the signal is emitted. That is, conditional on the current state, the probability distribution of the current signal is independent of all previous states visited by the process and all previous signals which have been emitted. For the signal emitted at the arbitrary time n, this can be expressed mathematically as follows:

$$P[S_n = v_k | X_1, S_1, \ldots, X_{n-1}, S_{n-1}, X_n = i] = P[S_n = v_k | X_n = i] = b_{ik} \tag{2.3}$$

for $i \in S,\ v_k \in \delta$ and $n \in \{1, 2, \ldots\}$.

By making use of equation (2.1), equation (2.3) can be extended to the following:

$$P[S_{n+t} = v_k | S_1, X_1, \ldots, S_n, X_n = i] = P[S_{n+t} = v_k | X_n = i] \tag{2.4}$$

where $t = 1, 2, 3 \ldots$.

An explanation of equation (2.4) is that at time $n+t$, conditional on all previous states and signals, only the state at time $n + t$ will determine the signal at time $n + t$ (by equation (2.3)). However, from equation (2.1) it can be seen that, given information up to time n, X_{n+t} is dependent only on X_n. Therefore, given information up to time n, S_{n+t} will also be dependent only on X_n.

Equation (2.4) can also be extended to the following:

$$\begin{aligned} &P[S_{k+t}, \ldots, S_n \,|\, S_1, \ldots, S_k, X_1, \ldots X_k] = P[S_{k+t}, \ldots, S_n \,|\, X_k] \\ &\text{where } k < n \text{ and } t = 1, 2, \ldots, n - k\,, \end{aligned} \tag{2.5}$$

$$\begin{aligned} &P[S_{k+t}, \ldots, S_n \,|\, S_1, \ldots, S_k, X_1, \ldots, X_n] = P[S_{k+t}, \ldots, S_n \,|\, X_{k+t}, \ldots, X_n] \\ &\text{where } k < n \text{ and } t = 1, 2, \ldots, n - k\,. \end{aligned} \tag{2.6}$$

These equations make intuitive sense and may be proven mathematically (see for example Appendix A of [35]).

Finally, since b_{ik} is a probability, the following must also hold:

$$\begin{aligned} &b_{ik} \geq 0, \text{ for } i \in S \text{ and } v_k \in \delta \\ &\sum_{v_k \in \delta} b_{ik} = 1, \text{ for } i \in S. \end{aligned} \tag{2.7}$$

2.2 Distribution Hidden Markov Models

In many applications of the HMM it may be convenient to assume that given the state which the HMM is in, the signal is emitted according to a familiar probability distribution - e.g. a binomial or a Poisson distribution. An example of this is a HMM whereby, given that the occupied state at some time point n is $i \in S$, the probability of observing the signal $x \in \{0, 1, 2, \ldots\}$ is governed by a Poisson distribution as follows[1]

$$P[S_n = x | X_n = i] = b_{ix} = \frac{e^{-\omega_i}\, \omega_i^x}{x!}\,. \tag{2.8}$$

That is, each state in the state space will emit a signal according to the particular Poisson distribution defined for that state.

Notice from the above that x is used to notate the observed signal for these HMMs since, depending on the signal distribution assumed, the signal space δ need not be discrete, as is implied by the usual notation ν_m. For example, in Section 4.1.4 of this dissertation the normal distribution is used to define the signal distribution for each state.

A variety of names are used in the literature to refer to such models; for example [46] refers to distribution HMMs (e.g. a Poisson HMM), [35] refers to hidden Markov time series models. This dissertation will make use of the term distribution HMMs to refer to these HMMs for which the signal distribution is specified by some parametric probability distribution.

An example of how a distribution HMM model can be used to improve the modelling of a series of data is given in [46]. In this example the series of annual counts of major earthquakes (magnitude 7 and above) for the years 1900-2006 is given. For ease of reference, Figure 2.1 plots this data. Since the observations are unbounded counts, the Poisson distribution would be a natural choice to describe them. How-

[1]Note that ω_i is used to notate the parameter of the Poisson distribution as λ, the usual notation, has been reserved to notate the parameter set of the HMM.

ever an examination of Figure 2.1 suggests that there may be some periods with a low rate of earthquakes, and some with a higher rate of earthquakes. This results in the earthquake series having a sample variance, $s^2 \approx 52$, which is much larger than the sample mean, $\bar{x} \approx 19$, indicating strong over-dispersion relative to the Poisson distribution (which has the property that the variance equals the mean). In addition to this, the sample autocorrelation function for this data, given in [46] - see page 29, suggests the presence of strong positive serial dependence. Hence a model consisting of independent Poisson random variables would be inappropriate due to the two reasons mentioned above.

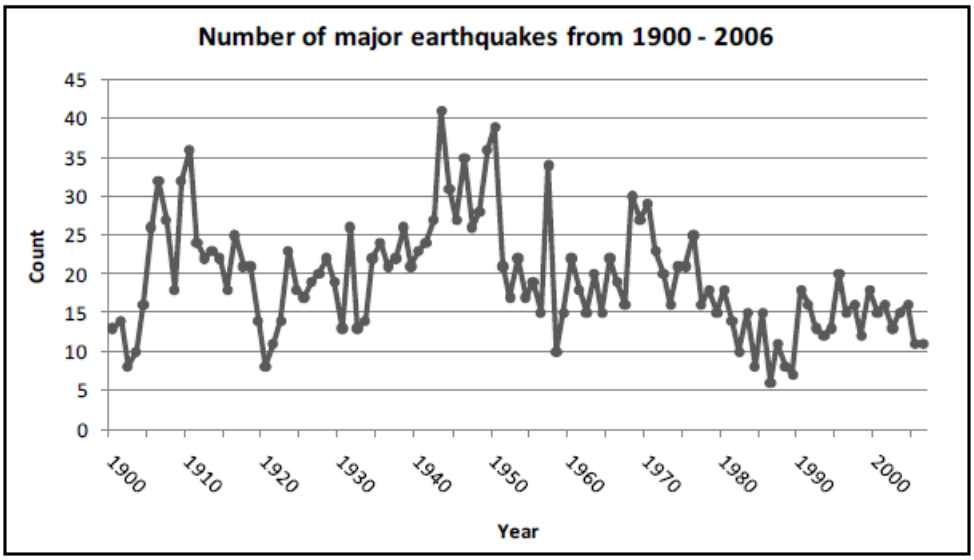


Figure 2.1: Number of major earthquakes in the world (magnitude 7 or greater), 1900-2006. This is based on data provided in [46], p. 4, Table 1.1.

One method of dealing with over-dispersed observations with a multimodal distribution is to use a mixture model. Mixture models are designed to accommodate unobserved heterogeneity in the population; that is mixture models will attempt to model unobserved groups in the population, with each group having a distinct distribution for the observed variable. In the above example, suppose that each count of the earthquake series is generated by one of two Poisson distributions, with means ω_1

and ω_2, where the choice of mean is determined by another process referred to as the parameter process. In the simple case, this parameter process is a series of independent random variables, giving rise to independent counts. Such a model is termed an independent mixture model. It is shown in [46] that if ω_1 is chosen with probability δ_1 and ω_2 is chosen with probability δ_2, then the variance in the resulting distribution exceeds the mean by $\delta_1\delta_2(\omega_1 - \omega_2)^2$. Hence an independent Poisson mixture model permits overdispersion, that is s^2 exceeding $\bar{x}$.

However, an independent Poisson mixture model is not ideal for the earthquake series as, by definition, it does not allow for serial dependence in the observations. One way of allowing for this serial dependence is to relax the assumption that the parameter process is serially independent. A mathematically convenient way to do this is to assume that the parameter process is a Markov chain. The resulting model is then an example of the Poisson HMM described above. Thus modelling the observed earthquake counts with a Poisson HMM will overcome the mentioned shortcomings which arise when a model consisting of independent Poisson random variables is used. The above example thus illustrates how a HMM can be interpreted as a mixture model which allows for serial dependence among the observations, thereby further highlighting the usefulness of the HMM. These are also properties of the general HMMs described in the previous section.

Appendix A of this dissertation proves that, as expected, the independent mixture model is indeed a special case of the HMM.

Finally, it is important to note that the HMM assumptions and properties described in Section 2.1 will also hold true for distribution HMMs. An overview on parameter estimation for distribution HMMs is provided in Section 4.1.4 of this dissertation. Further variations of distribution HMMs are also discussed in [35] and [46] - see for example pages 116 to 118 of [46].

2.3 Deriving Important Equations for the Hidden Markov Model

In order for HMMs to be useful in applications, three particular problems regarding the HMM need to be solved - namely the evaluation problem, the decoding problem and the learning problem. The solutions to these problems rely heavily on three equations - namely the forward, backward and Viterbi equations. While the solution to the three mentioned problems will be discussed further in Chapters 3 and 4, the three equations of interest needed to solve these problems will be defined and discussed in this section.

To begin, some elementary statistical results will be needed. The first of these results is the *partition rule for probability* which states:

If $\{B_r; r \geq 1\}$ forms a partition of the sample space Ω, that is $B_j \cap B_k = \phi$ for $j \neq k$, and $\bigcup_r B_r = \Omega$, then

$$P(A) = \sum_r P(A \cap B_r)\,. \tag{2.9}$$

Other elementary results which will be used during the derivation of the forward, backward and Viterbi equations are:

$$P(A \cap B) = P(B)P(A|B) \tag{2.10}$$

$$\begin{aligned} P(A \cap B|C) &= \frac{P\{A \cap (B \cap C)\}}{P(C)} \quad \text{.....} \quad \text{by (2.10)} \\ &= \frac{P(B \cap C)P(A|B \cap C)}{P(C)} \quad \text{.....} \quad \text{by (2.10)} \\ &= \frac{P(C)P(B|C)P(A|B \cap C)}{P(C)} \quad \text{.....} \quad \text{by (2.10)} \\ &= P(B|C)P(A|B \cap C)\,. \end{aligned} \tag{2.11}$$

The derivations provided in the next sections are adapted from those given in [41] and will make regular use of the results stated above.

2.3.1 Deriving the Forward Equation

This section will define and derive a computational form for the forward equation.

To begin, let

$$\mathbf{S}_n = (S_1, \ldots, S_n)$$

be a vector of random variables for the first n signals, and

$$\mathbf{s}_n = (s_1, \ldots, s_n)$$

be the actual sequence of the first n signals which have been observed, where $s_k \in \delta$ for $k = 1, 2 \ldots, n$.

Now, for $j \in S$, let the forward equation be defined as follows:

$$F_k(j) = P(\mathbf{S}_k = \mathbf{s}_k, X_k = j|\lambda). \tag{2.12}$$

For ease of readability, the derivations below will suppress the explicit conditioning on λ, i.e. $F_k(j) = P(\mathbf{S}_k = \mathbf{s}_k, X_k = j)$:

$$\begin{aligned}
F_k(j) &= P(\mathbf{S}_k = \mathbf{s}_k, X_k = j) \\
&= P(\mathbf{S}_{k-1} = \mathbf{s}_{k-1}, S_k = s_k, X_k = j) \\
&= \sum_{i \in S} P(\mathbf{S}_{k-1} = \mathbf{s}_{k-1}, X_{k-1} = i, S_k = s_k, X_k = j) \quad \ldots\ldots \quad \text{by (2.9)} \\
&= \sum_{i \in S} P(\mathbf{S}_{k-1} = \mathbf{s}_{k-1}, X_{k-1} = i) P(S_k = s_k, X_k = j|\mathbf{S}_{k-1} = \mathbf{s}_{k-1}, X_{k-1} = i) \\
&\qquad\qquad \ldots\ldots \quad \text{by (2.10)} \\
&= \sum_{i \in S} F_{k-1}(i) P(S_k = s_k, X_k = j|\mathbf{S}_{k-1} = \mathbf{s}_{k-1}, X_{k-1} = i) \quad \ldots\ldots \quad \text{by (2.12)} \\
&= \sum_{i \in S} F_{k-1}(i) P(X_k = j|\mathbf{S}_{k-1} = \mathbf{s}_{k-1}, X_{k-1} = i) \\
&\qquad \times P(S_k = s_k|\mathbf{S}_{k-1} = \mathbf{s}_{k-1}, X_{k-1} = i, X_k = j) \quad \ldots\ldots \quad \text{by (2.11)}
\end{aligned}$$

$$= \sum_{i \in S} F_{k-1}(i) P(X_k = j | X_{k-1} = i) P(S_k = s_k | X_k = j) \quad \text{..... by (2.1) and (2.3)}$$

$$= \sum_{i \in S} F_{k-1}(i)\, p_{ij}\, P(S_k = s_k | X_k = j). \tag{2.13}$$

By letting

$$b_{j,s_k} = P(S_k = s_k | X_k = j), \quad \text{where } s_k \in \delta \text{ and } j \in S, \tag{2.14}$$

equation (2.13) can be used to define the forward equation at time k as a function of the forward equations at time $k-1$ (where $k = 2, 3, \ldots, n$), as shown below:

$$F_k(j) = b_{j,s_k} \sum_{i \in S} F_{k-1}(i)\, p_{ij}\,. \tag{2.15}$$

Generating the forward equation for the first n signals, $F_n(j)$, is done as follows:

Starting with $k = 1$,

$$\begin{aligned} F_1(j) &= P(S_1 = s_1, X_1 = j | \lambda) && \text{..... by (2.12)} \\ &= P(X_1 = j | \lambda) P(S_1 = s_1 | X_1 = j, \lambda) && \text{..... by (2.11)} \\ &= p_j\, b_{j,s_1} && \end{aligned} \tag{2.16}$$

must be calculated for each state $j \in S$.

Equation (2.15) is then used recursively to calculate

$$\begin{aligned} F_2(j) &= P(\mathbf{S}_2 = \mathbf{s}_2, X_2 = j | \lambda) = b_{j,s_2} \sum_{i \in S} F_1(i) p_{ij}, \quad \text{for each } j \in S \\ F_3(j) &= P(\mathbf{S}_3 = \mathbf{s}_3, X_3 = j | \lambda) = b_{j,s_3} \sum_{i \in S} F_2(i) p_{ij}, \quad \text{for each } j \in S \\ &\vdots \\ F_n(j) &= P(\mathbf{S}_n = \mathbf{s}_n, X_n = j | \lambda) = b_{j,s_n} \sum_{i \in S} F_{n-1}(i) p_{ij}, \quad \text{for each } j \in S. \end{aligned}$$

2.3.2 Deriving the Backward Equation

This section will derive a computational form for the backward equation, which is defined for $i \in S$ as follows:

$$B_k(i) = P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n | X_k = i, \lambda)\,. \tag{2.17}$$

For ease of readability, the derivations below will suppress the explicit conditioning on λ, i.e. $B_k(i) = P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n | X_k = i)$:

$$\begin{aligned}
B_k(i) &= P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n | X_k = i) \\
&= \sum_{j \in S} P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n, X_{k+1} = j | X_k = i) \quad \text{..... by (2.9)} \\
&= \sum_{j \in S} P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n | X_k = i, X_{k+1} = j) P(X_{k+1} = j | X_k = i) \\
&\qquad\qquad \text{..... by (2.11)} \\
&= \sum_{j \in S} P(S_{k+1} = s_{k+1} | X_k = i, X_{k+1} = j) \\
&\qquad \times P(S_{k+2} = s_{k+2}, \ldots, S_n = s_n | S_{k+1} = s_{k+1}, X_k = i, X_{k+1} = j)\, p_{ij} \\
&\qquad\qquad \text{..... by (2.11)} \\
&= \sum_{j \in S} P(S_{k+1} = s_{k+1} | X_{k+1} = j) P(S_{k+2} = s_{k+2}, \ldots, S_n = s_n | X_{k+1} = j)\, p_{ij} \\
&\qquad\qquad \text{..... by (2.3) and (2.5)} \\
&= \sum_{j \in S} P(S_{k+1} = s_{k+1} | X_{k+1} = j)\, B_{k+1}(j)\, p_{ij}\,. \qquad (2.18)
\end{aligned}$$

Using the notation introduced in equation (2.14), equation (2.18) can be used to express the backward equation at time k as a function of the backward equations at time $k+1$ (where $k = 1, 2, \ldots, n-1$), as shown below:

$$B_k(i) = \sum_{j \in S} b_{j,s_{k+1}}\, B_{k+1}(j)\, p_{ij}\,. \tag{2.19}$$

Generating the backward equations $B_k(i)$ for $k = 1, 2, \ldots, n$ is done by working backwards in time as follows:

Starting with $k = n$, set

$$B_n(i) = 1. \tag{2.20}$$

This is done since $B_n(i)$ calls upon the $(n+1)^{th}$ observed signal, but only the first n signals have been observed. Setting $B_n(i)$ to 1 ensures that $0 \leq B_k(i) \leq 1$ for $k = 1, 2, \ldots, n$ and $i \in S$, which of course needs to hold true since $B_k(i)$ is a probability.

Equation (2.19) is then used recursively to calculate

$$\begin{aligned}
B_{n-1}(i) &= P(S_n = s_n | X_{n-1} = i, \lambda) = \sum_{j \in S} b_{j,s_n} B_n(j) \, p_{ij} = \sum_{j \in S} b_{j,s_n} \, p_{ij} && i \in S \\
B_{n-2}(i) &= P(S_{n-1} = s_{n-1}, S_n = s_n | X_{n-2} = i, \lambda) = \sum_{j \in S} b_{j,s_{n-1}} B_{n-1}(j) \, p_{ij} && i \in S \\
&\vdots \\
B_1(i) &= P(S_2 = s_2, \ldots, S_n = s_n | X_1 = i, \lambda) = \sum_{j \in S} b_{j,s_2} B_2(j) \, p_{ij} && i \in S.
\end{aligned}$$

As a final comment, it should be noted that, by equation (2.5),

$$P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n | S_k = s_k, X_k = i, \lambda) = P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n | X_k = i, \lambda) \, .$$

Therefore, the backward equation could also have been defined as

$$B_k(i) = P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n | S_k = s_k, X_k = i, \lambda) \, . \tag{2.21}$$

The form of the backward equation expressed in equation (2.21) will be called upon in Section 6.1.3 of this dissertation when the DCMM is discussed.

2.3.3 Deriving the Viterbi Equation

This section will detail the Viterbi Equation. To begin, define

$$\mathbf{X}_k = (X_1, \ldots, X_k)$$

to be a vector of random variables for the first k states visited by the HMM.

In Section 3.2, one of the problems of interest will be to find the sequence of states $(i_1, \ldots, i_n)$ which maximises $P\{\mathbf{X}_n = (i_1, \ldots, i_n) | \mathbf{S}_n = \mathbf{s}_n, \lambda\}$, where $i_k \in S$ for $k = 1, \ldots, n$.

In order to solve this, Section 3.2 will call upon the Viterbi equation, which is defined, for $k \in \{1, 2, \ldots, n\}$, as follows:

$$\begin{aligned} V_k(j) &= \max_{i_1,\ldots,i_{k-1}} P\{\mathbf{X}_{k-1} = (i_1, \ldots, i_{k-1}), X_k = j, \mathbf{S}_k = \mathbf{s}_k | \lambda\} \\ &\quad \text{where } i_h \in S \text{ for } h = 1, \ldots, k-1\,. \end{aligned} \tag{2.22}$$

For ease of readability, the derivations below will suppress the explicit conditioning on λ, i.e. $V_k(j) = \max_{i_1,\ldots,i_{k-1}} P\{\mathbf{X}_{k-1} = (i_1, \ldots, i_{k-1}), X_k = j, \mathbf{S}_k = \mathbf{s}_k\}$:

$$\begin{aligned} V_k(j) &= \max_{i_1,\ldots,i_{k-1}} P\{\mathbf{X}_{k-1} = (i_1, \ldots, i_{k-1}), X_k = j, \mathbf{S}_k = \mathbf{s}_k\} \\ &= \max_{i \in S} \max_{i_1,\ldots,i_{k-2}} P\{\mathbf{X}_{k-2} = (i_1, \ldots, i_{k-2}), X_{k-1} = i, X_k = j, \mathbf{S}_k = \mathbf{s}_k\} \\ &= \max_{i \in S} \max_{i_1,\ldots,i_{k-2}} P\{\mathbf{X}_{k-2} = (i_1, \ldots, i_{k-2}), X_{k-1} = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}, X_k = j, S_k = s_k\} \\ &= \max_{i \in S} \max_{i_1,\ldots,i_{k-2}} [P\{\mathbf{X}_{k-2} = (i_1, \ldots, i_{k-2}), X_{k-1} = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}\} \\ &\qquad \times P\{X_k = j, S_k = s_k | \mathbf{X}_{k-2} = (i_1, \ldots, i_{k-2}), X_{k-1} = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}\}] \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots\ldots \text{ by (2.10)} \end{aligned}$$

$$
\begin{aligned}
&= \max_{i\in S} \max_{i_1,\ldots,i_{k-2}} [P\{\mathbf{X}_{k-2} = (i_1,\ldots,i_{k-2}), X_{k-1} = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}\} \\
&\quad \times P\{X_k = j | \mathbf{X}_{k-2} = (i_1,\ldots,i_{k-2}), X_{k-1} = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}\} \\
&\quad \times P\{S_k = s_k | \mathbf{X}_{k-2} = (i_1,\ldots,i_{k-2}), X_{k-1} = i, X_k = j, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}\}] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots\ldots \text{ by (2.11)} \\
&= \max_{i\in S} \max_{i_1,\ldots,i_{k-2}} [P\{\mathbf{X}_{k-2} = (i_1,\ldots,i_{k-2}), X_{k-1} = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}\} \\
&\quad \times P\{X_k = j | X_{k-1} = i\} P\{S_k = s_k | X_k = j\}] \quad \ldots\ldots \text{ by (2.1) and (2.3)} \\
&= P\{S_k = s_k | X_k = j\} \\
&\quad \times \max_{i\in S} [p_{ij} \max_{i_1,\ldots,i_{k-2}} P\{\mathbf{X}_{k-2} = (i_1,\ldots,i_{k-2}), X_{k-1} = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}\}] \\
&= P\{S_k = s_k | X_k = j\} \max_{i\in S} p_{ij} V_{k-1}(i) \quad \ldots\ldots \text{ by (2.22).}
\end{aligned}
$$

Using the notation introduced in equation (2.14), the above equation expresses the Viterbi equation at time k as a function of the Viterbi equations at time $k-1$ (where $k = 2, 3, \ldots, n$), as shown below:

$$
V_k(j) = b_{j,s_k} \max_{i\in S} \{p_{ij} V_{k-1}(i)\}. \tag{2.23}
$$

The Viterbi equations are then calculated recursively beginning with $V_1(j)$ up to $V_n(j)$, for each $j \in S$. It is convenient to show these recursive calculations in Section 3.2.2 when the Viterbi equations will be called upon.

Chapter 3

Solving Problems Regarding the Hidden Markov Model

Application of the HMM requires that three problems of interest regarding the model be solved. These three problems are the evaluation problem, the decoding problem and the learning problem. Discussions and solutions of these three problems are provided in this chapter, and in doing so, the forward, backward and Viterbi equations discussed in the previous chapter are called upon.

As will be shown in this chapter, once solved, the evaluation, decoding and learning problems serve as powerful tools when using the HMM.

3.1 The Evaluation Problem

3.1.1 Describing the Evaluation Problem

The evaluation problem is described as follows:

Given the sequence of n signals which have been observed, $\mathbf{s}_n = (s_1, \ldots, s_n)$, and a HMM with $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$, the question of interest for the evaluation problem is how to calculate $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$, the probability that the observed sequence of signals was

generated by the HMM λ.

If the process is currently at time 0 and λ is known, the evaluation method can be used to calculate the probability that a certain sequence for the first n signals will be observed. For instance, returning to the example described in Section 2.1.1, it may well be of interest to calculate the probability that the first n products (signals) that the machine will produce will all be satisfactory.

As mentioned, this view of the evaluation problem can only be used if λ is known, or if good estimates of the components of λ are known. However, in most applications λ is not known and needs to be estimated using the sequence of signals which has been observed (this is discussed further in Section 4). In such instances the evaluation problem provides a method of evaluating how efficiently a given model describes the observed sequence of signals. That is $P(\mathbf{S}_n = \mathbf{s}_n|\hat{\lambda})$ is a measure of the likelihood of an estimate of λ.

This viewpoint is particularly useful if the components of λ are not fully known (as is typically the case) and there are potentially k plausible models, $\lambda^{(1)}, \ldots, \lambda^{(k)}$ which are thought to describe the HMM well. Having observed a signal sequence, the probability that this sequence was generated by $\lambda^{(i)}$ can then be calculated, using the evaluation method, for each $i = 1, \ldots, k$. In this way, the model that best describes the sequence of observed signals (i.e. the model that has the highest probability $P(\mathbf{S}_n = \mathbf{s}_n|\lambda^{(i)})$) can be selected from the k plausible models.

3.1.2 Solving the Evaluation Problem

This section will describe how the probability $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ can be calculated, thereby providing a solution to the evaluation problem. The derivations given in this section are adapted from [37] and [41].

To illustrate the usefulness of the forward and backward equations (described in the

previous chapter), a solution to the evaluation problem will first be provided without making use of these two equations. Thereafter, the forward and backward equations will be called upon to provide a simplified method of solving the evaluation problem.

To begin, one way of solving the evaluation problem (without making use of the forward and backward equations) is as follows:

$$
\begin{aligned}
& P(\mathbf{S}_n = \mathbf{s}_n|\lambda) \\
= \quad & \sum_{i_1 \in S} \cdots \sum_{i_n \in S} P(\mathbf{S}_n = \mathbf{s}_n, X_1 = i_1, \ldots, X_n = i_n|\lambda) \quad \text{.....} \quad \text{by (2.9)} \\
= \quad & \sum_{i_1 \in S} \cdots \sum_{i_n \in S} P(S_1 = s_1, \ldots, S_n = s_n|X_1 = i_1, \ldots, X_n = i_n, \lambda) \\
& \qquad \times P(X_1 = i_1, \ldots, X_n = i_n|\lambda) \quad \text{.....} \quad \text{by (2.11)} \\
= \quad & \sum_{i_1 \in S} \cdots \sum_{i_n \in S} P(S_2 = s_2, \ldots, S_n = s_n|S_1 = s_1, X_1 = i_1, \ldots, X_n = i_n, \lambda) \\
& \qquad \times P(S_1 = s_1|X_1 = i_1, \ldots, X_n = i_n, \lambda)\, P(X_1 = i_1|\lambda) \\
& \qquad \times P(X_2 = i_2, \ldots, X_n = i_n|X_1 = i_1, \lambda) \quad \text{.....} \quad \text{by (2.11)} \\
= \quad & \sum_{i_1 \in S} \cdots \sum_{i_n \in S} P(S_2 = s_2, \ldots, S_n = s_n|X_2 = i_2, \ldots, X_n = i_n, \lambda)\, P(S_1 = s_1|X_1 = i_1, \lambda) \\
& \qquad \times p_{i_1}\, P(X_2 = i_2|X_1 = i_1, \lambda)\, P(X_3 = i_3, \ldots, X_n = i_n|X_1 = i_1, X_2 = i_2, \lambda) \\
& \qquad \text{(by (2.6), (2.11) and the fact that conditional on the current state, the} \\
& \qquad \text{current signal is independent of any future state)} \\
& \vdots \\
= \quad & \sum_{i_1 \in S} \cdots \sum_{i_n \in S} b_{i_1,s_1} \ \cdots\ b_{i_n,s_n}\, p_{i_1}\, p_{i_1,i_2}\, p_{i_2,i_3} \ \cdots\ p_{i_{n-1},i_n} \qquad (3.1) \\
& \qquad \text{(by repeatedly using (2.6), (2.11),} \\
& \qquad \text{the fact that conditional on the current state, the current signal} \\
& \qquad \text{is independent of any future state,} \\
& \qquad \text{and the fact that the state process possess the Markov property).}
\end{aligned}
$$

If there are N states in the state space, the above calculation would involve the summation of N^n terms, with each term being the product of $2n$ values. As this calculation can become very involved if either N or n is large, a more compact calculation for the evaluation method is desirable. To this end, computationally simpler calculations of $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ can be obtained by using either the forward equation, or the backward equation or a combination of the two.

Starting with the forward equation, the evaluation problem can be solved as follows:

$$\begin{aligned} P(\mathbf{S}_n = \mathbf{s}_n|\lambda) &= \sum_{i\in S} P(\mathbf{S}_n = \mathbf{s}_n, X_n = i|\lambda) \quad \text{..... by (2.9)} \\ &= \sum_{i\in S} F_n(i)\,, \end{aligned} \tag{3.2}$$

where the last step is obtained using equation (2.12), and $F_n(i)$ is obtained recursively (as described in Section 2.3.1) for each $i \in S$.

Alternatively, by using the backward equation, the evaluation problem can be solved in the following way:

$$\begin{aligned} P(\mathbf{S}_n = \mathbf{s}_n|\lambda) &= \sum_{i\in S} P(S_1 = s_1, \ldots, S_n = s_n, X_1 = i|\lambda) \quad \text{..... by (2.9)} \\ &= \sum_{i\in S} P(S_1 = s_1, \ldots, S_n = s_n|X_1 = i, \lambda) P(X_1 = i|\lambda) \quad \text{..... by (2.11)} \\ &= \sum_{i\in S} P(S_1 = s_1|X_1 = i, \lambda) P(S_2 = s_2, \ldots, S_n = s_n|S_1 = s_1, X_1 = i, \lambda)\, p_i \\ &\qquad\qquad \text{..... by (2.11)} \\ &= \sum_{i\in S} b_{i,s_1}\, P(S_2 = s_2, \ldots, S_n = s_n|X_1 = i, \lambda)\, p_i \quad \text{..... by (2.5)} \\ &= \sum_{i\in S} b_{i,s_1}\, B_1(i)\, p_i\,, \end{aligned} \tag{3.3}$$

where the last step is obtained using equation (2.17), and $B_1(i)$ is obtained recursively (as described in Section 2.3.2) for each $i \in S$.

If there are N states in the state space, either of the above two calculations would

involve calculating nN quantities, with each of the last $(n-1)N$ of these quantities $(F_2(i), \ldots, F_n(i)$ or $B_{n-1}(i), \ldots, B_1(i)$, for $i = 1, 2, \ldots, N)$ requiring a summation over N terms. This is a far more economical calculation for $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ than the previous approach described, thereby highlighting the importance of the forward and backward equations.

A further approach to calculate the evaluation method is to combine the forward and backward equations as follows:

$$\begin{aligned}
& P(\mathbf{S}_n = \mathbf{s}_n, X_k = i|\lambda) \\
= \; & P(\mathbf{S}_k = \mathbf{s}_k, S_{k+1} = s_{k+1}, \ldots, S_n = s_n, X_k = i|\lambda) \\
= \; & P(\mathbf{S}_k = \mathbf{s}_k, X_k = i|\lambda) \\
& \quad \times P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n|\mathbf{S}_k = \mathbf{s}_k, X_k = i, \lambda) \quad \ldots\ldots \quad \text{by (2.11)} \\
= \; & P(\mathbf{S}_k = \mathbf{s}_k, X_k = i|\lambda) \\
& \quad \times P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n|X_k = i, \lambda) \quad \ldots\ldots \quad \text{by (2.5)} \\
= \; & F_k(i)B_k(i)\,. \qquad (3.4)
\end{aligned}$$

It then follows that,

$$\begin{aligned}
P(\mathbf{S}_n = \mathbf{s}_n|\lambda) &= \sum_{i \in S} P(\mathbf{S}_n = \mathbf{s}_n, X_k = i|\lambda)) \quad \ldots\ldots \quad \text{by (2.9)} \\
&= \sum_{i \in S} F_k(i)B_k(i)\,. \qquad (3.5)
\end{aligned}$$

It should be noted that if $k = n$ in the above, then by equation (2.20), equation (3.5) is equivalent to (3.2). Similarly, by equation (2.16), if $k = 1$ then equation (3.5) is equivalent to (3.3).

Using the above, $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ can be determined by recursively using the forward and backward equations to calculate $F_k(i)$ from $F_1(i)$ and $B_k(i)$ from $B_n(i)$, for each $i \in S$. These computations can then be stopped once both $F_k(i)$ and $B_k(i)$ have been calculated for each $i \in S$.

3.1.3 Illustrating the Evaluation Problem

This section aims to further supplement the evaluation method by revisiting the example provided in Section 2.1.1 to explicitly illustrate the calculations required to perform the evaluation method. These calculations, which are an expansion of what is given in [41], will be based on the **P** and **B** matrices given in Section 2.1.2, and the vector of initial probabilities $\mathbf{a} = \begin{bmatrix} 0.8 & 0.2 \end{bmatrix}$. In addition, suppose that the first three items produced by the machine are 'satisfactory', 'defective' and 'satisfactory'. Therefore, using the notation from Section 2.1.2, the observed signal sequence for the first 3 signals will be $\mathbf{s}_3 = (\nu_1, \nu_2, \nu_1)$.

The forward equations are recursively calculated as follows:

$$\begin{aligned}
F_1(1) &= (0.8)(0.99) = 0.7920 \\
F_1(2) &= (0.2)(0.96) = 0.1920 \\
\\
F_2(1) &= (0.01)\{(0.7920)(0.9) + (0.1920)(0)\} = 0.0071 \\
F_2(2) &= (0.04)\{(0.7920)(0.1) + (0.1920)(1)\} = 0.0108 \\
\\
F_3(1) &= (0.99)\{(0.0071)(0.9) + (0.0108)(0)\} = 0.0064 \\
F_3(2) &= (0.96)\{(0.0071)(0.1) + (0.0108)(1)\} = 0.0111\,.
\end{aligned}$$

The calculations for the backward equations are as follows:

$$\begin{aligned}
B_3(1) &= 1 \\
B_3(2) &= 1 \\
\\
B_2(1) &= (0.99)(1)(0.9) + (0.96)(1)(0.1) = 0.9870 \\
B_2(2) &= (0.99)(1)(0) + (0.96)(1)(1) = 0.9600 \\
\\
B_1(1) &= (0.01)(0.9870)(0.9) + (0.04)(0.9600)(0.1) = 0.0127 \\
B_1(2) &= (0.01)(0.9870)(0) + (0.04)(0.9600)(1) = 0.0384\,.
\end{aligned}$$

In Section 3.1.2 three approaches using the forward and backward equations to solve the evaluation problem were given (see equations (3.2), (3.3) and (3.5)). For completeness, all 3 of these methods are shown below:

$$\begin{aligned} P(\mathbf{S}_3 = \mathbf{s}_3|\lambda) &= \sum_{i=1}^{2} F_3(i) = 0.0064 + 0.0111 = 0.0174 \qquad \text{or,} \\ P(\mathbf{S}_3 = \mathbf{s}_3|\lambda) &= \sum_{i=1}^{2} b_{i,1}\, B_1(i)\, p_i \quad (\text{recall that } b_{i,1} = P(S_k = \nu_1|X_k = i)) \\ &= (0.99)(0.0127)(0.8) + (0.96)(0.0384)(0.2) \\ &= 0.0174 \qquad \text{or,} \\ P(\mathbf{S}_3 = \mathbf{s}_3|\lambda) &= \sum_{i=1}^{2} F_2(i) B_2(i) \\ &= (0.0071)(0.9870) + (0.0108)(0.9600) \\ &= 0.0174\,. \end{aligned}$$

If the first three items produced were all 'satisfactory' - denoted as $\mathbf{s}_3^* = (\nu_1, \nu_1, \nu_1)$, then $P(\mathbf{S}_3 = \mathbf{s}_3^*|\lambda)$ is calculated, using the same methodology as above, to be $0.9464\,$.

These probabilities can be interpreted in one of two ways.

If the signal sequence has already been observed, then the evaluation probability can be viewed as the probability that the HMM λ produced the signal sequence. If the signal sequence which was observed is $\mathbf{s}_3$, there would be little confidence that the parameters specified for λ are accurate. However, an observed signal sequence of $\mathbf{s}_3^*$ would provide more comfort that the parameters of λ are accurately specified.

The second way the probabilities from the evaluation method can be interpreted is as follows. If there is confidence in the accuracy of the specified HMM λ, and if the signal sequence has not yet been observed, then the probability that the first three items produced will be 'satisfactory', 'defective' and 'satisfactory' is 1.74%; while the probability that the first three items produced will all be 'satisfactory' is $94.64\%\,$.

Based on these calculations, the machine owners can have a view regarding the quality to expect for the first three products.

3.2 The Decoding Problem

3.2.1 Describing the Decoding Problem

Given a HMM with parameter set λ and a sequence of n signals which have been observed, $\mathbf{s}_n = (s_1, \ldots, s_n)$, the decoding problem entails finding the most likely sequence for the n states which have been visited by the model. Solving the decoding problem is then a matter of unravelling the hidden part of the model.

Since the sequence of states produced by the model is at no point observable, no 'correct' state sequence can be found. Instead, optimisation techniques are used to maximise likelihood functions, thereby finding an optimal sequence of states which best describes the sequence of signals which has been observed. Unlike the evaluation problem, more than one approach to solve the decoding problem is possible which could potentially lead to solutions which are not necessarily the same (i.e. different estimated state sequences). In Section 3.2.2, two possible approaches are contrasted.

3.2.2 Solving the Decoding Problem

This section will discuss the derivations of two possible methods which can be used to solve the decoding problem. These approaches have been adapted from [33], [37], [38] and [41].

To begin, let

$$\hat{X}_k \text{ denote the optimal estimator of } X_k \in S.$$

As has been the case in previous sections, assume that the first n signals have been observed and that this observed signal sequence is denoted by $\mathbf{s}_n = (s_1, \ldots, s_n)$.

The task of this section is then to find $\hat{X}_k$ for each $k = 1, 2, \ldots, n$.

The first optimisation technique looks to maximise the number of individual states which are correctly predicted. This entails finding the state at each time point which is individually the most probable given the sequence of n observed signals; that is, finding the state $i \in S$ which maximises the likelihood

$$P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda),$$

for each $k = 1, 2, \ldots, n$.

Described mathematically, $\hat{X}_k$ is derived as follows for this optimisation technique:

$$\hat{X}_k = \arg\max_{i \in S} P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda), \qquad \text{for each } k = 1, 2, \ldots, n.$$

Now, notice that

$$\begin{aligned} P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) &= \frac{P(\mathbf{S}_n = \mathbf{s}_n, X_k = i | \lambda)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \quad \text{..... by (2.11)} \\ &= \frac{F_k(i) B_k(i)}{\sum_{j \in S} F_k(j) B_k(j)}, \end{aligned} \tag{3.6}$$

where the last step of the above was obtained using equations (3.4) and (3.5).

By noting that $\sum_{j \in S} F_k(j) B_k(j)$ is constant for each $i \in S$, it follows that given $\mathbf{S}_n = \mathbf{s}_n$,

$$\hat{X}_k = \arg\max_{i \in S} \{F_k(i) B_k(i)\} \tag{3.7}$$

for each $k = 1, 2, \ldots, n$.

While (3.7) maximises the number of individually correct states, the 'optimal' state sequence produced by (3.7) may not always result in an attainable state sequence. To see this, suppose that the HMM has a zero state transition probability between the two states i and j (for some $i, j \in S$), that is $p_{ij} = 0$. This technique cannot guarantee that if $\hat{X}_k = i$ then $\hat{X}_{k+1} \neq j$ for all $k = 1, 2, \ldots, (n-1)$ in the 'optimal'

state sequence. Similarly, $p_{ij} = 1$ (for some $i, j \in S$) may also result in an impossible 'optimal' state sequence if this technique is used.

The above problem occurs due to the fact that the solution in (3.7) determines the most probable state individually at every time point without regarding the probability of state sequences as a whole. However, the solution given in (3.7) can at times still prove useful as these problematic situations (where $p_{ij} = 0$ or $p_{ij} = 1$) will not always occur in practice. Also this technique is computationally quite simple to perform once the forward and backward equations have been calculated for each time point.

A solution to the decoding problem which overcomes the shortfalls of the above method would be appealing. To this end, the most widely used technique in the field of HMMs is to regard the entire state sequence as a single entity. Using this approach, the solution to the decoding problem will be the state sequence $(i_1, \ldots, i_n)$ which maximises

$$P(\mathbf{X}_n = (i_1, \ldots, i_n) | \mathbf{S}_n = \mathbf{s}_n, \lambda) \,,$$

a likelihood containing the entire joint state sequence.

Now, by equation (2.11), the following holds:

$$P(\mathbf{X}_n = (i_1, \ldots, i_n) | \mathbf{S}_n = \mathbf{s}_n, \lambda) = \frac{P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \,. \tag{3.8}$$

Since the calculation of $P(\mathbf{S}_n = \mathbf{s}_n | \lambda)$ does not depend on the state sequence which has been visited (see Section 3.1), the problem of interest is equivalent to finding the state sequence $(i_1, \ldots, i_n)$ which will maximise

$$P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda) \,. \tag{3.9}$$

In other words, the optimal state sequence for this approach is defined as

$$(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n) = \arg \max_{i_1, \ldots, i_n} P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda)$$

where $i_k \in S$ for each $k = 1, 2, \ldots, n$.

For a given state sequence, the derivation of equation (3.1) in Section 3.1.2 showed that

$$P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n|\lambda) = b_{i_1,s_1} \ \ldots \ b_{i_n,s_n} \, p_{i_1} \, p_{i_1,i_2} \, p_{i_2,i_3} \ \ldots \ p_{i_{n-1},i_n} \, . \quad (3.10)$$

Using this, the likelihood expressed in (3.9) can be calculated for each possible state sequence, and in so doing, the optimal state sequence (that is, the state sequence which maximises (3.9)) can be found. If there are N states in the state space, the total number of possible state sequences is N^n. This number will grow rapidly if either n or N is large and so, for this reason, a less computationally intensive approach is required.

To this end, typically the ***Viterbi algorithm*** is used for this purpose.

The Viterbi algorithm (VA) is defined in [24] as a "recursive optimal solution to the problem of estimating the state sequence of a discrete-time finite-state Markov process observed in memoryless noise". The VA then seems well suited to finding the hidden state sequence of a HMM as this state sequence is a Markov process while the signal sequence, conditional on the state sequence, is memoryless. The algorithm however is by no means restricted to the field of the HMM. In fact, the VA was originally proposed in [44] as an algorithm for decoding convolutional codes and, as mentioned in [25], has since been extended and is used in numerous applications within the fields of (amongst others) decoding, communications (such as deep-space communication, mobile communication and digital video broadcasting) and of course HMMs. Within the field of HMMs it has been widely used in a variety of pattern recognition problems, particularly for speech recognition (see for example [37]) and computational biology (where the VA is used to locate genes in DNA sequences - see for example [29]). While the VA extends beyond the field of HMMs, in order to keep the research presented in this dissertation relevant, further discussions of the VA will be within the context of the HMM.

Returning to the topic of interest for this section, recall that, in order to solve the

decoding problem, it is required that the VA be called upon to determine the state sequence which will maximise (3.9). A description on how the algorithm is performed will be provided next. Thereafter it will be proven that the state sequence which is determined by the VA is indeed the optimal state sequence which maximises (3.9). It will also be shown that this optimal state sequence overcomes the problem which was experienced when (3.7) was used to determine the optimal state sequence (that being that the 'optimal' state sequence may in actual fact be an unattainable state sequence).

To begin, for $k \in \{2, 3, \ldots, n\}$, let

$$V_k(j) = \max_{i_1,\ldots,i_{k-1}} P\{\mathbf{X}_{k-1} = (i_1, \ldots, i_{k-1}), X_k = j, \mathbf{S}_k = \mathbf{s}_k|\lambda\}\,. \tag{3.11}$$

This equation is none other than the Viterbi equation which was discussed in Section 2.3.3 (see equation (2.22)). In equation (2.23) it is shown that

$$V_k(j) = b_{j,s_k} \max_{i \in S} \{p_{ij}\, V_{k-1}(i)\}\,. \tag{3.12}$$

Now if $k = 1$ then, for each $j \in S$, equation (3.12) is reduced to

$$\begin{aligned} V_1(j) &= P(X_1 = j, S_1 = s_1|\lambda) \\ &= P(X_1 = j|\lambda)P(S_1 = s_1|X_1 = j, \lambda) \quad \text{.....} \quad \text{by (2.11)} \\ &= p_j\, b_{j,s_1}\,. \end{aligned} \tag{3.13}$$

In order to perform the VA, for $k \in \{2, 3, \ldots, n\}$, let

$\psi_k(j)$ be the state that maximised $\{p_{ij}\, V_{k-1}(i)\}$ in the calculation of $V_k(j)$.

That is, let

$$\psi_k(j) = \arg\max_{i \in S} \{p_{ij}\, V_{k-1}(i)\}.$$

The VA is then performed by recursively working forward, using equations (3.12) and (3.13), to calculate the following:

$$
\begin{aligned}
V_1(j) &= p_j\, b_{j,s_1} && \text{for each } j \in S \\
V_2(j) &= b_{j,s_2} \max_{i \in S} \{p_{ij}\, V_1(i)\} && \text{for each } j \in S \\
\psi_2(j) &= \arg\max_{i \in S} \{p_{ij}\, V_1(i)\} && \text{for each } j \in S \\
V_3(j) &= b_{j,s_3} \max_{i \in S} \{p_{ij}\, V_2(i)\} && \text{for each } j \in S \\
\psi_3(j) &= \arg\max_{i \in S} \{p_{ij}\, V_2(i)\} && \text{for each } j \in S \\
&\vdots \\
V_n(j) &= b_{j,s_n} \max_{i \in S} \{p_{ij}\, V_{n-1}(i)\} && \text{for each } j \in S \\
\psi_n(j) &= \arg\max_{i \in S} \{p_{ij}\, V_{n-1}(i)\} && \text{for each } j \in S.
\end{aligned}
$$

For a given k and j, there is a possibility in the above that $\psi_k(j)$ has more than one value. If this occurs, then there exists multiple optimal state sequences which will result in equivalent likelihood values.

Finding the optimal state sequence is then done by recursively working backwards through the above as follows:

$$
\begin{aligned}
\hat{X}_n &= \arg\max_{j \in S} \{V_n(j)\} \\
\hat{X}_{n-1} &= \psi_n(\hat{X}_n) \\
\hat{X}_{n-2} &= \psi_{n-1}(\hat{X}_{n-1}) \\
&\vdots \\
\hat{X}_2 &= \psi_3(\hat{X}_3) \\
\hat{X}_1 &= \psi_2(\hat{X}_2)\,. \qquad (3.14)
\end{aligned}
$$

The optimal state sequence, as determined by the Viterbi Algorithm, will then be $(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n)$.

The above process can be explained as follows.

Firstly define the likelihood score to be the value of likelihood (3.9) for a given state sequence. Now, for each state in the state space, the initial likelihood score (that is the value of the likelihood if $n = 1$) is calculated.

For a given state $j \in S$, at time 2 one must then find the most likely transition to j and update the likelihood score (for when $n = 2$) for this transition. This is done by multiplying the transition probabilities of all the transitions coming into state j with their corresponding previous likelihood scores, and selecting the transition with the maximum product to be the most likely transition to j. This most likely transition to state j, at time 2, would then be $(\psi_2(j), j)$.

The likelihood score for the most likely transition to state j at time 2 is then obtained by multiplying the above maximum product by b_{j,s_2} . This updated likelihood score is none other than $V_2(j)$ - which makes sense since, by definition, $V_2(j) = \max_{i_1 \in S} P\{X_1 = i_1, X_2 = j, \mathbf{S}_2 = \mathbf{s}_2 | \lambda\}$.

The above is repeated for all $j \in S$. This process is then performed recursively until time n is reached. In this way, for each time point $k = 2, 3, \ldots, n$ and each state $j \in S$, $\psi_k(j)$ - the state which will give rise to the most likely transition into state j at time k - is determined.

At the end of the recursive process, the final state in the optimal state sequence $(\hat{X}_n)$ is found by examining which state at time n gives rise to the maximum likelihood score at time n. The optimal state at time $n-1$ $(\hat{X}_{n-1})$ is then chosen to be the state which was determined as the most likely to transition into $\hat{X}_n$ at time n. This state was of course determined during the forward recursions as $\psi_n(\hat{X}_n)$.

Similarly, the optimal state at time $n-2$ is then chosen to be $\psi_{n-1}(\hat{X}_{n-1})$, the state which was determined during the forward recursions as the most likely to transition into $\hat{X}_{n-1}$ at time $n-1$. This process is continued until the optimal state at time 1 is determined, thereby giving rise to the optimal state sequence as determined by the VA.

The above has detailed the procedure of performing the VA in the context of HMMs. Three theorems are presented next to further expand the VA. As the basis of these theorems are only outlined in literature (see for example [41]), the formal proofs to these theorems have been explicitly derived by the author of this dissertation. These are presented below.

To begin, a question of interest regarding the VA is whether it achieves the objective initially desired - that is, does the VA find the state sequence which will maximise (3.9). This is indeed the case, as is formally proven in Theorem 1 below.

Theorem 1: The optimal state sequence derived from the Viterbi Algorithm is the state sequence which maximises the likelihood

$$P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda)\,,$$

where $i_k \in S$ for each $k = 1, 2, \ldots, n$.

Proof: Firstly assume that $(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n)$ is the state sequence which is generated by the VA (defined in equation (3.14)).

Now, by (3.10), for a given state sequence $(i_1, \ldots, i_n)$, the likelihood of interest can be calculated as follows

$$P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda) = b_{i_1,s_1} \ \ldots \ b_{i_n,s_n}\, p_{i_1}\, p_{i_1,i_2} \ \ldots \ p_{i_{n-1},i_n}\,.$$

Thus, if the VA does indeed maximise the likelihood of interest, then the following will hold

$$\max_{i_1,\ldots,i_n} P\{\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda\} = b_{\hat{X}_1,s_1} \ \ldots \ b_{\hat{X}_n,s_n}\, p_{\hat{X}_1}\, p_{\hat{X}_1,\hat{X}_2} \ \ldots \ p_{\hat{X}_{n-1},\hat{X}_n}\,. \tag{3.15}$$

To prove that this is the case, consider the following:

$$
\begin{aligned}
& \max_{i_1,\ldots,i_n} P\{\mathbf{X}_n = (i_1,\ldots,i_n), \mathbf{S}_n = \mathbf{s}_n|\lambda\} \\
= \quad & \max_{j\in S} \max_{i_1,\ldots,i_{n-1}} P\{\mathbf{X}_{n-1} = (i_1,\ldots,i_{n-1}), X_n = j, \mathbf{S}_n = \mathbf{s}_n|\lambda\} \\
= \quad & \max_{j\in S} V_n(j) \quad \text{.....} \quad \text{by (3.11)} \\
= \quad & V_n(\hat{X}_n) \quad \text{.....} \quad \text{since, by (3.14), } \hat{X}_n = \arg\max_{j\in S}\{V_n(j)\} \\
= \quad & b_{\hat{X}_n,s_n} \max_{i\in S}\{p_{i,\hat{X}_n} V_{n-1}(i)\} \quad \text{.....} \quad \text{by (3.12)} \\
= \quad & b_{\hat{X}_n,s_n} p_{\psi_n(\hat{X}_n),\hat{X}_n} V_{n-1}(\psi_n(\hat{X}_n)) \\
& \qquad (\text{since } \psi_n(j) = \arg\max_{i\in S}\{p_{ij} V_{n-1}(i)\}, \text{it follows that} \\
& \qquad\quad \psi_n(\hat{X}_n) = \arg\max_{i\in S}\{p_{i,\hat{X}_n} V_{n-1}(i)\}) \\
= \quad & b_{\hat{X}_n,s_n} p_{\hat{X}_{n-1},\hat{X}_n} V_{n-1}(\hat{X}_{n-1}) \quad \text{.....} \quad \text{by (3.14)} \\
= \quad & b_{\hat{X}_n,s_n} p_{\hat{X}_{n-1},\hat{X}_n} b_{\hat{X}_{n-1},s_{n-1}} \max_{i\in S}\{p_{i,\hat{X}_{n-1}} V_{n-2}(i)\} \quad \text{.....} \quad \text{by (3.12)} \\
= \quad & b_{\hat{X}_n,s_n} p_{\hat{X}_{n-1},\hat{X}_n} b_{\hat{X}_{n-1},s_{n-1}} p_{\psi_{n-1}(\hat{X}_{n-1}),\hat{X}_{n-1}} V_{n-2}(\psi_{n-1}(\hat{X}_{n-1})) \\
& \qquad (\text{since } \psi_{n-1}(\hat{X}_{n-1}) = \arg\max_{i\in S}\{p_{i,\hat{X}_{n-1}} V_{n-2}(i)\}) \\
= \quad & b_{\hat{X}_n,s_n} p_{\hat{X}_{n-1},\hat{X}_n} b_{\hat{X}_{n-1},s_{n-1}} p_{\hat{X}_{n-2},\hat{X}_{n-1}} V_{n-2}(\hat{X}_{n-2}) \quad \text{.....} \quad \text{by (3.14)} \\
& \vdots \\
= \quad & b_{\hat{X}_n,s_n} \ldots b_{\hat{X}_2,s_2} p_{\hat{X}_{n-1},\hat{X}_n} \ldots p_{\hat{X}_1,\hat{X}_2} V_1(\hat{X}_1) \\
= \quad & b_{\hat{X}_n,s_n} \ldots b_{\hat{X}_2,s_2} b_{\hat{X}_1,s_1} p_{\hat{X}_{n-1},\hat{X}_n} \ldots p_{\hat{X}_1,\hat{X}_2} p_{\hat{X}_1} \quad \text{.....} \quad \text{by (3.13).}
\end{aligned}
$$

Thus, equation (3.15) holds - thereby proving that the state sequence which is derived from the Viterbi Algorithm is indeed the state sequence which maximises the likelihood $P(\mathbf{X}_n = (i_1,\ldots,i_n), \mathbf{S}_n = \mathbf{s}_n|\lambda)$.

The final comments which need to be made are those surrounding the attainability of the optimal state sequence calculated using the VA.

Recall, from earlier in the section, that it cannot be guaranteed that the 'optimal' state sequence calculated by (3.7) will indeed be an attainable state sequence. The optimal state sequence produced by the VA will however always be an attainable one, as is proven by the two theorems below.

Theorem 2: If $(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n)$ is the optimal state sequence determined by the VA, $p_{i,j} = 0$ (for some $i, j \in S$) and $\hat{X}_k = i$ (for some $k = 1, 2, \ldots, n-1$), then $\hat{X}_{k+1} \neq j$.

Proof: Recall that the underlying assumption for this section (which was stated at the beginning of the section) is that the first n signals generated by the HMM λ have been observed.

Now, since a single signal is generated each time the HMM enters a new state, the HMM must have followed some state path to have generated the signal sequence which has been observed. That is, there must exist a least one state sequence $(i_1, \ldots, i_n)$ such that $P(\mathbf{X}_n = (i_1, \ldots, i_n) | \mathbf{S}_n = \mathbf{s}_n, \lambda) > 0$, where $i_1, i_2, \ldots, i_n \in S$.

It then follows, by equation (3.8), that $P\{\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda\} > 0$ for at least one state sequence $(i_1, \ldots, i_n)$. Thus,

$$\max_{i_1,\ldots,i_n} P\{\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda\} > 0\,. \tag{3.16}$$

Since $(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n)$ is the optimal state sequence determined by the VA, equation (3.15) holds true (validated in the proof of Theorem 1), that is

$$\max_{i_1,\ldots,i_n} P\{\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda\} = b_{\hat{X}_1,s_1} \ldots b_{\hat{X}_n,s_n} p_{\hat{X}_1} p_{\hat{X}_1,\hat{X}_2} \ldots p_{\hat{X}_{n-1},\hat{X}_n}.$$

Now, assume that $\hat{X}_{k+1} = j$. Then, from the hypothesis of the theorem, it follows that $p_{\hat{X}_k,\hat{X}_{k+1}} = p_{i,j} = 0$ for some $k = 1, 2, \ldots, n-1$. This then implies,

from the above equation, that

$$\max_{i_1,\ldots,i_n} P\{\mathbf{X}_n = (i_1,\ldots,i_n), \mathbf{S}_n = \mathbf{s}_n|\lambda\} = 0\,.$$

This however results in a contradiction of equation (3.16). Therefore the assumption that $\hat{X}_{k+1} = j$ must be incorrect.

Hence $\hat{X}_{k+1} \neq j$, thereby completing the proof.

Theorem 3: If $(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n)$ is the optimal state sequence determined by the VA, $p_{i,j} = 1$ (for some $i, j \in S$) and $\hat{X}_k = i$ (for some $k = 1, 2, \ldots, n-1$), then $\hat{X}_{k+1} = j\,.$

Proof: Using the same arguments as was used in the proof for Theorem 2, equation (3.16) can once again be established. Also, as was stated in the proof for Theorem 2, since $(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n)$ is the optimal state sequence determined by the VA, equation (3.15) holds true.

Now, assume that $\hat{X}_{k+1} \neq j\,.$ Then, from the hypothesis of the theorem, it follows that $p_{\hat{X}_k, \hat{X}_{k+1}} = 0$ for some $k = 1, 2, \ldots, n-1\,.$ This, by equation (3.15), implies that

$$\max_{i_1,\ldots,i_n} P\{\mathbf{X}_n = (i_1,\ldots,i_n), \mathbf{S}_n = \mathbf{s}_n|\lambda\} = 0\,.$$

This however results in a contradiction of equation (3.16). Therefore the assumption that $\hat{X}_{k+1} \neq j$ must be incorrect.

Hence, $\hat{X}_{k+1} = j\,,$ thereby completing the proof.

The above two theorems show that the VA will always calculate an attainable optimal state sequence.

3.2.3 Illustrating the Decoding Problem

An illustrative example showing calculations for the decoding problem is provided in this section. As the Viterbi algorithm is better illustrated in a 3 state HMM than by

the 2 state HMM provided in example 2.1.1, consider the HMM (adapted from [40]) $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$, where

$$\mathbf{P} = \begin{pmatrix} 0.8 & 0.05 & 0.15 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \\ 0.3 & 0.7 \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}.$$

Further suppose that 3 signals have been observed, producing the observed signal sequence $\mathbf{s}_3 = (\nu_2, \nu_1, \nu_1)$.

The calculations illustrating the VA, which are an expansion of those in [40], are as follows:

For $k = 1$,

$$\begin{aligned} V_1(1) &= p_1\, b_{12} = \frac{1}{3}(0.9) = 0.300 \\ V_1(2) &= p_2\, b_{22} = \frac{1}{3}(0.2) = 0.067 \\ V_1(3) &= p_3\, b_{32} = \frac{1}{3}(0.7) = 0.233. \end{aligned}$$

For $k = 2$,

$$\begin{aligned} V_2(1) &= (b_{11}) \max_{i \in S}\{p_{i1} V_1(i)\} = (0.1) \max\{(0.8)(0.300), (0.2)(0.067), (0.2)(0.233)\} \\ &= (0.1) \max\{0.2400, 0.0133, 0.0467\} = (0.1)(0.2400) = 0.0240 \\ \Psi_2(1) &= \arg\max_{i \in S}\{p_{i1}\, V_1(i)\} = 1 \\ V_2(2) &= (b_{21}) \max_{i \in S}\{p_{i2} V_1(i)\} = (0.8) \max\{((0.05)(0.300), (0.6)(0.067), (0.3)(0.233)\} \\ &= (0.8) \max\{0.0150, 0.0400, 0.0700\} = (0.8)(0.0700) = 0.0560 \\ \Psi_2(2) &= \arg\max_{i \in S}\{p_{i2}\, V_1(i)\} = 3 \\ V_2(3) &= (b_{31}) \max_{i \in S}\{p_{i3} V_1(i)\} = (0.3) \max\{((0.15)(0.300), (0.2)(0.067), (0.5)(0.233)\} \\ &= (0.3) \max\{0.0450, 0.0133, 0.1167\} = (0.3)(0.1167) = 0.0350 \\ \Psi_2(3) &= \arg\max_{i \in S}\{p_{i3}\, V_1(i)\} = 3. \end{aligned}$$

For $k = 3$,

$$\begin{aligned}
V_3(1) &= (b_{11}) \max_{i \in S}\{p_{i1} V_2(i)\} = (0.1) \max\{(0.8)(0.0240), (0.2)(0.0560), (0.2)(0.0350)\} \\
&= (0.1) \max\{0.0192, 0.0112, 0.0070\} = (0.1)(0.0192) = 0.0019 \\
\Psi_3(1) &= \arg\max_{i \in S}\{p_{i1}\, V_2(i)\} = 1 \\
V_3(2) &= (b_{21}) \max_{i \in S}\{p_{i2} V_2(i)\} = (0.8) \max\{(0.05)(0.0240), (0.6)(0.0560), (0.3)(0.0350)\} \\
&= (0.8) \max\{0.0012, 0.0336, 0.0105\} = (0.8)(0.0336) = 0.0269 \\
\Psi_3(2) &= \arg\max_{i \in S}\{p_{i2}\, V_2(i)\} = 2 \\
V_3(3) &= (b_{31}) \max_{i \in S}\{p_{i3} V_2(i)\} = (0.3) \max\{(0.15)(0.0240), (0.2)(0.0560), (0.5)(0.0350)\} \\
&= (0.3) \max\{0.0036, 0.0112, 0.0175\} = (0.3)(0.0175) = 0.0053 \\
\Psi_3(3) &= \arg\max_{i \in S}\{p_{i3}\, V_2(i)\} = 3.
\end{aligned}$$

Working backwards using the above,

$$\begin{aligned}
\hat{X}_3 &= \arg\max_{j \in S}\{V_3(j)\} = \arg\max\{0.0019, 0.0269, 0.0053\} = 2 \\
\hat{X}_2 &= \Psi_3(\hat{X}_3) = \Psi_3(2) = 2 \\
\hat{X}_1 &= \Psi_2(\hat{X}_2) = \Psi_2(2) = 3.
\end{aligned}$$

And so the most likely state sequence, as determined by the Viterbi Algorithm, is $(3, 2, 2)$.

It was proven in Section 3.2.2 that this is the state sequence which will maximise the likelihood (3.9) for the observed signal sequence $\mathbf{s}_3 = (\nu_2, \nu_1, \nu_1)$. The actual value of the likelihood is calculated using equation (3.10) as follows

$$\begin{aligned}
P(\mathbf{X}_3 = (3, 2, 2), \mathbf{S}_3 = \mathbf{s}_3 | \lambda) &= (b_{32})\,(b_{21})\,(b_{21})\,(p_3)\,(p_{32})\,(p_{22}) \\
&= (0.7)\,(0.8)\,(0.8)\,(1/3)\,(0.3)\,(0.6) \\
&= 0.0269\,. \qquad (3.17)
\end{aligned}$$

At first glance this probability may seem low. Recall however that there are 27 possible state sequences and 8 possible signal sequences on which the above probability is defined. Due to this large set of possible values, it should not be unexpected that the probability of an individual state sequence and an individual signal sequence occurring be low.

Conditional on the signal sequence which has been observed, the probability that the Viterbi state sequence is indeed the state sequence which has visited is calculated, using equation (3.8), as follows

$$\begin{aligned} P(\mathbf{X}_3 = (3,2,2)|\mathbf{S}_3 = \mathbf{s}_3, \lambda) &= \frac{P(\mathbf{X}_3 = (3,2,2), \mathbf{S}_3 = \mathbf{s}_3|\lambda)}{P(\mathbf{S}_3 = \mathbf{s}_3|\lambda)} \\ &= \frac{0.0269}{0.0825} \\ &= 0.3259, \end{aligned}$$

where $P(\mathbf{S}_3 = \mathbf{s}_3|\lambda)$ was calculated using the evaluation method. As expected, this conditional probability is notably higher than the probability in equation (3.17).

Recall from Section 3.2.2 that the 'optimal' state sequence can also be determined by maximising the number of *individual* states which are correctly predicted, that is

$$\hat{X}_k = \arg\max_{i \in S} P(X_k = i|\mathbf{S}_3 = \mathbf{s}_3, \lambda)\,, \qquad \text{for each } k = 1, 2, 3\,.$$

By deriving the forward and backward equations, and by making use of equation (3.7), the optimal state sequence for this optimisation technique is calculated to be (3,2,2). For this example, this optimal state sequence is equivalent to the optimal state sequence obtained through the Viterbi Algorithm.

3.3 The Learning Problem

3.3.1 Describing the Learning Problem

In most applications, the values for the parameters of a given HMM, $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$, are unknown and therefore need to be estimated. The purpose of the learning problem is then to determine an appropriate methodology to optimally estimate these unknown parameters for the HMM. This is typically done by finding the parameter set which will maximise some likelihood of the observed signal sequence. That is, the learning method looks to optimally estimate the model parameters of λ so as to best describe the signal sequence which has been observed.

Depending on the application, different model specifications may be considered for the HMM. While the same broad principals may be used to estimate the model parameters, these may need modifying depending on the specified HMM. As such, variations of solving the learning problem may need to be considered according to the model specification of the HMM which is assumed.

Due to the rich literature available on this topic, details on how the learning problem can be resolved will be covered in the next chapter. For convenience however, a brief summary of this is given below.

For the general case of the time homogeneous, discrete-time, discrete-state and discrete-signal HMM which has been discussed in the previous sections of this dissertation, an iterative algorithm known as the Baum-Welch algorithm (BWA) is commonly referenced by the literature as a solution to the learning problem. This algorithm will be detailed in Section 4.1.

In certain applications of the HMM it may be convenient to assume that signals are emitted according to a familiar probability distribution (for a example a Poisson distribution or a binomial distribution). These HMMs were introduced as distribution HMM in Section 2.2. As a result of the differing signal distributions, the classical

BWA must be adapted in order to estimate the unknown signal probabilities for these distribution HMMs. An overview regarding this is provided in Section 4.1.4.

An alternative approach (to the BWA), of optimally estimating the parameters for the HMM is introduced in Section 4.2. This approach looks to maximise the likelihood function directly through the use of numerical techniques.

In summary, since in most applications of the HMM the exact values for the parameters of λ are unknown, the learning method is crucial as it looks to optimally estimate these unknown parameters based on the signal sequence which has been observed.

3.4 Other Statistical Properties of Interest

Additional statistical properties of the HMM may be of interest in many applications. These include marginal distributions and moments, as well as calculating the probabilities that certain states will be visited at future time points and that certain signals will be emitted at future time points. The work presented in this section is adapted predominantly from [41] and [46].

3.4.1 Marginal Distributions

Since it is assumed that the underlying hidden state process of a HMM is a Markov chain (which is in no way driven by the signals which are emitted), the marginal distribution for the states will follow the results which were given in Section 1.4. That is, the marginal probability $p_i(k) = P(X_k = i)$, for $i \in S$ and $k \in \{1, 2, \ldots, n\}$, can be calculated from the initial state probabilities, $\mathbf{a} = (p_1, p_2, \ldots, p_i, \ldots)$, and the one-step state transition probabilities, $\mathbf{P}$, by making use of equation (1.8). Of course, if the underlying Markov chain is stationary then the marginal distribution for the states is simply $p_i(k) = \pi_i$ for each $i \in S$.

The marginal distribution for S_k, the signal emitted at time $k \in \{1, 2, \ldots, n\}$, is as

follows

$$\begin{aligned} P(S_k = \nu_m) &= \sum_{i \in S} P(X_k = i) P(S_k = \nu_m | X_k = i) \quad \text{.....} \quad \text{by (2.9) and (2.10)} \\ &= \sum_{i \in S} p_i(k)\, b_{im} \end{aligned} \tag{3.18}$$

for each $\nu_m \in \delta$.

In some applications, the bivariate marginal distributions may be required. These can be derived by firstly noting that the joint distribution of a set of random variables V_i is given by

$$P(V_1, V_2, \ldots, V_z) = \prod_{i=1}^{Z} P(V_i \,|\, \text{pa}(V_i))\,,$$

where $\text{pa}(V_i)$ denotes all the 'parents' of V_i in the set $V_1, V_2, \ldots, V_z$ (i.e., the variables on which V_i is dependant on); see for example [19], p. 250.

Considering the four random variables S_k, S_{k+h}, X_k and X_{k+h} for positive integer h, it can be seen that $\text{pa}(X_k)$ is empty, $\text{pa}(S_k) = \{X_k\}$, $\text{pa}(X_{k+h}) = \{X_k\}$ and $\text{pa}(S_{k+h}) = \{X_{k+h}\}$. It therefore follows that

$$P(S_k, S_{k+h}, X_k, X_{k+h}) = P(X_k)\, P(S_k|X_k)\, P(X_{k+h}|X_k)\, P(S_{k+h}|X_{k+h}).$$

Hence

$$\begin{aligned} & P(S_k = \nu_p\,,\, S_{k+h} = \nu_q) \\ = & \sum_{i \in S} \sum_{j \in S} P(S_k = \nu_p, S_{k+h} = \nu_q, X_k = i, X_{k+h} = j) \\ = & \sum_{i \in S} \sum_{j \in S} P(X_k = i)\, P(S_k = \nu_p \,|\, X_k = i)\, P(X_{k+h} = j \,|\, X_k = i)\, P(S_{k+h} = \nu_q \,|\, X_{k+h} = j) \\ = & \sum_{i \in S} \sum_{j \in S} p_i(k)\, b_{ip}\, p_{ij}(h)\, b_{jq}\,, \end{aligned}$$

where $\mathbf{P}^{(h)} = \{p_{ij}(h)\} = \mathbf{P}^h$ by equation (1.6).

Similarly, $P(X_k = i, X_{k+h} = j) = p_i(k) p_{ij}(h)$.

Expressions for higher-order marginal distributions can similarly be derived.

3.4.2 Moments

For certain applications of the HMM it may be necessary to calculate moments for either the state or signal for a given time point. This is easy enough for the general HMM described thus far as the marginal distributions are known and can be computed as shown in Section 3.4.1.

For the case of the distribution HMMs described in Section 2.2, it is often convenient to use the following to express the moments for the signal emitted at time k:

$$\mathrm{E}(S_k) = \sum_{i \in S} \mathrm{E}(S_k \,|\, X_k = i)\, P(X_k = i) \;=\; \sum_{i \in S} p_i(k)\, \mathrm{E}(S_k \,|\, X_k = i)\,.$$

More generally, the following analogous results hold:

$$\begin{aligned} \mathrm{E}(g(S_k)) &= \sum_{i \in S} p_i(k)\, \mathrm{E}(g(S_k) \,|\, X_k = i) \\ \mathrm{E}(g(S_k, S_{k+h})) &= \sum_{i \in S} \sum_{j \in S} \mathrm{E}(g(S_k, S_{k+h}) \,|\, X_k = i, X_{k+h} = j)\, p_i(k)\, p_{ij}(h)\,. \end{aligned}$$

So, for example, if it is assumed that the signal of a HMM is emitted from state i by the (univariate) state-dependent distribution $b_i(s)$, and that μ_i and σ_i^2 denote the mean and variance of the distribution b_i, then it can easily be verified using the above (see [46] for details) that:

$$\begin{aligned} \mathrm{E}(S_k) &= \sum_{i \in S} p_i(k)\, \mu_i \\ \mathrm{E}(S_k^2) &= \sum_{i \in S} p_i(k)\, (\sigma_i^2 + \mu_i^2) \\ \mathrm{Var}(S_k) &= \left[\sum_{i \in S} p_i(k)\, (\sigma_i^2 + \mu_i^2)\right] - \left[\sum_{i \in S} p_i(k)\, \mu_i\right]^2 \\ \mathrm{E}(S_k, S_{k+h}) &= \sum_{i \in S} \sum_{j \in S} \mu_i\, \mu_j\, p_i(k)\, p_{ij}(h)\,. \end{aligned}$$

Using the above, $\mathrm{Cov}(S_k, S_{k+h})$ and $\mathrm{Corr}(S_k, S_{k+h})$ can also easily be calculated.

3.4.3 Forecasting Future States and Signals

Forecasting distributions for both the signals and states of the general HMM described thus far are derived in this section.

To begin, firstly assume that n signals have been observed. Now, notice that for each $j \in S$

$$\begin{aligned} P(X_n = j|\mathbf{S}_n = \mathbf{s}_n, \lambda) &= \frac{P(\mathbf{S}_n = \mathbf{s}_n, X_n = j|\lambda)}{P(\mathbf{S}_n = \mathbf{s}_n|\lambda)} \quad \text{..... by (2.11)} \\ &= \frac{F_n(j)}{\sum_{l \in S} F_n(l)} \quad \text{..... by (2.12) and (3.2).} \end{aligned}$$

This is consistent with equation (3.6) in Section 3.2.2 since, by definition, $B_n(i) = 1$ for each $i \in S$.

The forecasting distribution for the state visited at time $n+h$, where positive integer h is termed the forecast horizon, can be derived as follows:

$$\begin{aligned} & P(X_{n+h} = j|\mathbf{S}_n = \mathbf{s}_n, \lambda) \\ = & \sum_{i \in S} P(X_{n+h} = j, X_n = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{..... by (2.9)} \\ = & \sum_{i \in S} P(X_{n+h} = j|X_n = i, \mathbf{S}_n = \mathbf{s}_n, \lambda)\, P(X_n = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{..... by (2.11)} \\ = & \sum_{i \in S} P(X_{n+h} = j|X_n = i, \lambda)\, P(X_n = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{..... by (2.1)} \\ = & \frac{1}{\sum_{l \in S} F_n(l)} \sum_{i \in S} p_{ij}(h)\, F_n(i)\,. \end{aligned} \tag{3.19}$$

The forecasting distribution for the signal emitted at time $n + h$ can be derived as follows:

$$
\begin{aligned}
& P(S_{n+h} = \nu_m | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
= & \sum_{i \in S} P(S_{n+h} = \nu_m, X_{n+h} = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.9)} \\
= & \sum_{i \in S} P(S_{n+h} = \nu_m | X_{n+h} = i, \mathbf{S}_n = \mathbf{s}_n, \lambda)\, P(X_{n+h} = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.11)} \\
= & \sum_{i \in S} P(S_{n+h} = \nu_m | X_{n+h} = i, \lambda)\, P(X_{n+h} = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.3)} \\
= & \sum_{i \in S} b_{im}\, P(X_{n+h} = i | \mathbf{S}_n = \mathbf{s}_n, \lambda)\,, \qquad (3.20)
\end{aligned}
$$

where $P(X_{n+h} = i | \mathbf{S}_n = \mathbf{s}_n, \lambda)$ can be calculated using the equation (3.19) for each $i \in S$.

Chapter 4

Solving the Learning Problem for the Hidden Markov Model

This chapter will review how the learning problem can be solved for the HMM. Recall from Section 3.3 that this involves estimating unknown parameters for a given HMM.

To begin, parameter estimation for the Markov Chain is briefly discussed. In particular, if there is a relatively lengthy history of transitions available, then statistical inference for the transition probabilities can be made. For example, the maximum likelihood estimate of p_{ij} is given (see [1]) by:

$$\hat{p}_{ij} \quad = \quad \sum_{t=2}^{T} n_{ij}(t) \,/\, \sum_{t=1}^{T-1} n_i(t) \tag{4.1}$$

where $n_{ij}(t)$ is the number individuals which are in state i at time $t-1$ and state j at time t,

$n_i(t)$ is the number of individuals which are in state i at time t,

T is the last time point of available historical observations, and

$t = 1$ is the time at which the process begins.

And so a relatively simple analytical solution exists to find the maximum likelihood estimate of p_{ij} for the Markov chain. Section 4.1 presents an analytical approach

for estimating the parameters of the time homogeneous, discrete-time, discrete-state and discrete-signal HMM discussed in the previous sections of this dissertation. This approach, also based on likelihood maximisation, is an iterative algorithm which is commonly referenced in the literature as the Baum-Welch algorithm (BWA). As shall be seen in Section 4.1, while the BWA does indeed provide an analytical solution to parameter estimation for the HMM, this solution is considerably more complex than the direct analytic parameter estimators for the Markov chain (equation (4.1)).

4.1 The Baum-Welch Algorithm

4.1.1 Describing the Baum-Welch Algorithm

Before detailing the BWA, a brief historical overview (adapted from [35]) is provided. The algorithm was developed through a series of papers ([5], [6], [7], [8] and [9]) published by L.E. Baum and his co-workers between 1966 and 1972. The name Welch seems to just appear as the joint author (with Baum) of a paper referenced only within [9]. The algorithm is in fact an early example of the Expectation Maximization (EM) algorithm (a description of the EM algorithm and the relationship of the BWA to the EM algorithm are provided in Appendix B). It should be noted that some references in the literature refer to the BWA as the forward-backward algorithm (since, as will be shown, the previously defined forward and backward equations form part of the algorithm).

The focus of this section will be to explain how the BWA is performed and to give an intuitive overview as to why the algorithm works. Details of several implementation considerations which should be taken into account when performing the algorithm are also given in this section. These discussions are adapted primarily from [35] and [37]. Rigorous mathematics supporting the algorithm and an explanation as to how the BWA fits into the EM framework is provided in Appendix B. Since the mathematics of this relationship is often overlooked or only briefly accounted for in the

literature, it is believed that the work presented in this appendix adds definite value to the existing HMM literature.

The BWA can be described as follows: given the sequence of the first n observed signals, $\mathbf{s}_n = (s_1, ..., s_n)$, the BWA looks to estimate the HMM parameters, $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$, such that the likelihood $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ is maximised. The estimators calculated from the BWA are thus maximum likelihood estimators.

Before detailing the algorithm, some additional equations and their computational forms are required.

For $k = 1, ..., n-1$, notice that

$$\begin{aligned}
& P(\mathbf{S}_n = \mathbf{s}_n, X_k = i, X_{k+1} = j|\lambda) \\
= \; & P(\mathbf{S}_k = \mathbf{s}_k, X_k = i, X_{k+1} = j, S_{k+1} = s_{k+1}, ..., S_n = s_n|\lambda) \\
= \; & P(\mathbf{S}_k = \mathbf{s}_k, X_k = i|\lambda)\, P(X_{k+1} = j, S_{k+1} = s_{k+1}, ..., S_n = s_n|\mathbf{S}_k = \mathbf{s}_k, X_k = i, \lambda) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{.....} \quad \text{by (2.11)} \\
= \; & P(\mathbf{S}_k = \mathbf{s}_k, X_k = i|\lambda)\, P(X_{k+1} = j|\mathbf{S}_k = \mathbf{s}_k, X_k = i, \lambda) \\
& \quad \times P(S_{k+1} = s_{k+1}, ..., S_n = s_n|\mathbf{S}_k = \mathbf{s}_k, X_k = i, X_{k+1} = j, \lambda) \quad \text{.....} \quad \text{by (2.11)} \\
= \; & F_k(i)\, p_{ij}\, P(S_{k+1} = s_{k+1}, ..., S_n = s_n|X_{k+1} = j, \lambda) \quad \text{.....} \quad \text{by (2.1), (2.6), (2.12)} \\
= \; & F_k(i)\, p_{ij}\, P(S_{k+1} = s_{k+1}|X_{k+1} = j, \lambda) \\
& \quad \times P(S_{k+2} = s_{k+2}, ..., S_n = s_n|X_{k+1} = j, S_{k+1} = s_{k+1}, \lambda) \quad \text{.....} \quad \text{by (2.11)} \\
= \; & F_k(i)\, p_{ij}\, b_{j,s_{k+1}}\, P(S_{k+2} = s_{k+2}, ..., S_n = s_n|X_{k+1} = j, \lambda) \quad \text{.....} \quad \text{by (2.5)} \\
= \; & F_k(i)\, p_{ij}\, b_{j,s_{k+1}}\, B_{k+1}(j)\,. \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (4.2)
\end{aligned}$$

For $i, j \in S$ and $\nu_m \in \delta$, define

$$\begin{aligned}
\xi_k(i,j) &= P(X_k = i, X_{k+1} = j | \mathbf{S}_n = \mathbf{s}_n, \lambda)\,, \quad \text{and} \\
\gamma_k(i) &= P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
\gamma_{k,m}(i) &= \begin{cases} \gamma_k(i) & \text{if} \quad s_k = \nu_m \\ 0 & \text{if} \quad s_k \neq \nu_m\,. \end{cases}
\end{aligned} \tag{4.3}$$

Computational forms for $\xi_k(i,j)$ and $\gamma_k(i)$ are be derived as follows:

$$\begin{aligned}
\xi_k(i,j) &= P(X_k = i, X_{k+1} = j | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \frac{P(\mathbf{S}_n = \mathbf{s}_n, X_k = i, X_{k+1} = j | \lambda)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \quad \text{.....} \quad \text{by (2.11)} \\
&= \frac{F_k(i)\, p_{ij}\, b_{j,s_{k+1}}\, B_{k+1}(j)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \quad \text{.....} \quad \text{by (4.2)}
\end{aligned}$$

$$\begin{aligned}
\gamma_k(i) &= P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \frac{F_k(i) B_k(i)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \quad \text{.....} \quad \text{by (3.6).}
\end{aligned} \tag{4.4}$$

The probability $P(\mathbf{S}_n = \mathbf{s}_n | \lambda)$ in the above may be calculated using any of the representations of the evaluation calculation which were obtained in Section 3.1.2.

The following interpretations can be made about the above probabilities. While these results are intuitive, they are also formally proven in Appendix B:

$\sum_{k=1}^{n} \gamma_k(i)$ = expected number of times the HMM is in state i during the first n observed time points,

$\sum_{k=1}^{n-1} \gamma_k(i)$ = expected number of transitions from state i during the first n observed time points,

$$\begin{aligned}
\sum_{k=1}^{n-1} \xi_k(i,j) &= \text{expected number of transitions from state } i \text{ into state } j \text{ during the} \\
&\quad \text{first } n \text{ observed time points,} \\
\sum_{k=1}^{n} \gamma_{k,m}(i) &= \text{expected number of times the HMM is in state } i \text{ and emits signal } \nu_m \\
&\quad \text{during the first } n \text{ observed time points.}
\end{aligned} \tag{4.5}$$

As the BWA is an iterative algorithm, define $\lambda^* = (\mathbf{P}^*, \mathbf{B}^*, \mathbf{a}^*)$ to be the current estimate of the parameters for the HMM, and $\hat{\lambda} = (\hat{\mathbf{P}}, \hat{\mathbf{B}}, \hat{\mathbf{a}})$ to be the re-estimate of λ^*.

Also define

$$\gamma_k^*(i),\ \xi_k^*(i,j) \text{ and } \gamma_{k,m}^*(i) \tag{4.6}$$

to be the values for $\gamma_k(i),\ \xi_k(i,j)$ and $\gamma_{k,m}(i)$ calculated using λ^*.

Then for $i, j \in S$ and $\nu_m \in \delta$ the elements of $\hat{\lambda}$ can be calculated as follows:

$$\begin{aligned}
\hat{p}_i &= P(X_1 = i | \mathbf{S}_n = \mathbf{s}_n, \lambda^*) \\
&= \gamma_1^*(i)
\end{aligned} \tag{4.7}$$

$$\begin{aligned}
\hat{p}_{ij} &= \text{proportion of times that, when the HMM is in state } i, \text{ a transition into state } j \text{ occurs} \\
&= \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \\
&= \frac{\sum_{k=1}^{n-1} \xi_k^*(i,j)}{\sum_{k=1}^{n-1} \gamma_k^*(i)}
\end{aligned} \tag{4.8}$$

$$
\begin{aligned}
\hat{b}_{jm} &= \text{proportion of times that, when the HMM is in state } j\text{, signal } \nu_m \text{ is emitted} \\
&= \frac{\text{expected number of times the process is in state } j \text{ and emits signal } \nu_m}{\text{expected number of times the process is in state } j} \\
&= \frac{\sum_{k=1}^{n} \gamma^*_{k,m}(j)}{\sum_{k=1}^{n} \gamma^*_{k}(j)} \,. \qquad (4.9)
\end{aligned}
$$

The expressions (4.7)-(4.9), evaluated at the current parameter estimates, provide iteratively updated estimates of p_i, p_{ij} and b_{jm}.

As was mentioned in the introductory paragraphs of this section, the BWA is in fact an example of the Expectation Maximization (EM) algorithm; that is the Baum-Welch re-estimation equations (equations (4.7)-(4.9)) are identical to the iteration steps of the EM algorithm applied to this particular problem. The mathematics showing this are presented in Appendix B. This relationship is important as it allows conclusions regarding the properties of the BWA estimates to be made - as is highlighted in the paragraphs below.

An important result regarding the BWA is given next. This result is a property of estimates which are derived from the EM algorithm and is thus inherited by the BWA. The result, also proven in [8], states that for the BWA either:

1) λ^* defines a critical value of the likelihood function, $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$, in which case the above calculations will produce $\hat{\lambda} = \lambda^*$, or

2) model $\hat{\lambda}$ results in a higher value in the likelihood function than λ^* - that is $P(\mathbf{S}_n = \mathbf{s}_n|\hat{\lambda}) > P(\mathbf{S}_n = \mathbf{s}_n|\lambda^*)$. Therefore a new model, $\hat{\lambda}$, has been found from which the observed signal sequence is more likely to have been produced.

Based on the above findings, if $\hat{\lambda}$ is iteratively used in place of λ^* in the re-estimation calculations, the probability of the observed signal sequence being produced by the estimated model is improved until convergence is achieved. The final result of this

re-estimation procedure is then the maximum likelihood estimator. It should however be noted that the BWA only leads to a local maxima of the likelihood function, and that in most applications many local maxima are likely to exist. This however is the best which can be done since, to the best of knowledge at the time of writing, no analytical or numerical methods exist which will solve for the global maxima of the likelihood $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$. Since only a local maxima can be found, the choice of the initial values of λ used for the BWA will influence the final estimated values.

A pleasing property of the Baum-Welch re-estimation procedure is that at each iteration the following constraints of the HMM are met (provided of course that the initial estimates chosen for the BWA satisfy these constraints):

$$\begin{aligned}
&\sum_{i\in S} \hat{p}_i = 1 \\
&\hat{p}_i \geqslant 0, \qquad\qquad \text{for } i \in S, \text{ and} \\
&\sum_{j\in S} \hat{p}_{ij} = 1, \qquad \text{for } i \in S \\
&\hat{p}_{ij} \geqslant 0, \qquad\qquad \text{for } i, j \in S, \text{ and} \\
&\sum_{v_k\in\delta} \hat{b}_{ik} = 1, \qquad \text{for } i \in S\,. \\
&\hat{b}_{ik} \geq 0, \qquad\qquad \text{for } i \in S \text{ and } v_k \in \delta.
\end{aligned} \tag{4.10}$$

So, based on the above, the final estimated values of λ produced by the BWA will satisfy the HMM constraints given in equations (1.2), (1.3) and (2.7).

The property that $\hat{p}_i \geqslant 0$, $\hat{p}_{ij} \geqslant 0$ and $\hat{b}_{jk} \geqslant 0$ for each iteration is guaranteed from the fact that forward and backward equations will be guaranteed to be greater than or equal to zero for each observed time point, provided that the initial estimates of $\hat{p}_i$, $\hat{p}_{ij}$ and $\hat{b}_{jk}$ are chosen to be greater than or equal to zero (see Section 2.3 for details of this).

The remaining three properties of equation (4.10) can once again be proven by con-

sidering how the Baum-Welch re-estimation equations can be derived by making use of the EM algorithm. It is shown in Appendix B that when deriving the re-estimation equations which will maximise $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$, Lagrange multipliers are used in the EM algorithm to ensure that these constraints are satisfied for each iteration.

Incidentally, these three properties can also be proven algebraically by noting that if the partition rule for probability (equation (2.9)) is applied to equation (4.3), then $\sum_{j\in S} \xi_k(i,j) = \gamma_k(i)$ and $\sum_{\nu_m\in\delta} \gamma_{k,m}(j) = \gamma_k(j)$ is obtained, which when applied to equations (4.7)-(4.9) yield the desired properties.

Some further remarks regarding the BWA for HMMs are made below.

Initial probabilities estimated by the Baum-Welch algorithm

Recall that the vector $\mathbf{a}$ contains the initial state probabilities at time 1; that is $\mathbf{a}$ will contain $p_i = P(X_1 = i)$ for each $i \in S$.

Implementation of the BWA reveals that at a maximum of the likelihood, $\hat{p}_i$ will tend to 1 for some $i \in S$ and $\hat{p}_j$ will tend to 0 for the remaining $j \in S$. That is, if there are m states in the state space, the value for $\mathbf{a}$ which will maximise the likelihood (and therefore be estimated by the BWA) will tend to one of the m possible unit vectors. This is also noted in [31] and [34] - see page 1055 and 305 respectively.

For certain applications of the HMM, this however may not be acceptable or intuitively correct. In such instances an approach used in the literature is to fix the initial state probabilities at a pre-determined value and then use the BWA equations (equations (4.8) to (4.9)) to determine the state transition and signal probabilities which will maximise the likelihood under these fixed initial state probabilities. This could be done for a range of initial state probability values, with the final estimates for λ being the parameter set which yields the highest likelihood value.

The Baum-Welch algorithm for stationary HMMs

In some applications of the HMM it may be necessary to assume that the underlying Markov chain is stationary (see equation (1.9)). Under this assumption however, the BWA equations for the $\hat{p}_i$ and $\hat{p}_{ij}$ parameters (equations (4.7) to (4.8)) will no longer hold. Practically, this can be seen from the fact that the BWA estimate of $\mathbf{a}$ (equation (4.7)) is a unit vector (as discussed above). As such, the BWA estimates for $\hat{p}_i$ and $\hat{p}_{ij}$ given in equations (4.7) to (4.8) will not respect stationarity.

It is shown in Appendix B (see Section B.2.2) that due to the additional constraint implied by stationarity[1], a different function needs to be maximised when performing the EM algorithm - see equation (B.10). Analytical maximisation of (B.10) becomes rather involved, even for a two state HMM (as is shown in [46]). It is therefore suggested in [15] and [46] that, under the assumption of stationarity, numerical techniques be incorporated into the BWA to maximise (B.10).

An alternative to using the BWA, that being direct maximisation of the likelihood function, is discussed in Section 4.2.

4.1.2 Implementation Considerations for the Baum-Welch Algorithm

The methodology required to perform the BWA was detailed in the previous section. This section discusses implementation considerations which should be taken into account when applying the BWA.

4.1.2.1 Scaling

In order to understand why scaling may be necessary for the implementation of the BWA, consider the calculation of the forward equation, $F_k(j)$ (equation (2.15)). Since

[1]This constraint is $\mathbf{a} = \mathbf{1}(\mathbf{I}_m - \mathbf{P} + \mathbf{U}_m)^{-1}$ and was discussed in Section 1.4 (see equation (1.10)).

the calculation of $F_k(j)$ is an iterative procedure whereby the terms which are multiplied are all less than 1 (generally significantly less than 1), as k becomes large (that is, the signal sequence becomes long), $F_k(j)$ tends exponentially to 0. Similar comments hold for the computations of the backward equations. This fact is evident even in the simple example which was given in Section 2.1.1 where the forward and backward equations for a signal sequence of length 3 already started to tend towards 0 (see Section 3.1.3). Depending on the software used to perform the BWA, for significantly large k, the dynamic range of the $F_k(j)$ and $B_k(j)$ computations may exceed the precision range of the software.

Since both the forward and backward equations are required for the implementation of the BWA, the incorporation of a scaling procedure for the calculation of the BWA re-estimation equations may clearly be needed. The goal of this scaling procedure is then to ensure that $F_k(j)$ and $B_k(j)$ are kept within the dynamic range of the computer (for $k = 1, \ldots, n$), while ensuring that the re-estimation equations of the Baum-Welch algorithm still produce the same outcome.

A basic scaling procedure which achieves this, adapted from [37], is the following:

To begin, let

$$\begin{aligned} &F_k^{(s)}(j) \text{ denote the scaled version of } F_k(j), \text{ and} \\ &B_k^{(s)}(j) \text{ denote the scaled version of } B_k(j)\,. \end{aligned}$$

Now define, for each $j \in S$ and $k = 1, \ldots, n$

$$F_k^{(s)}(j) = \frac{\tilde{F}_k(j)}{\sum_{i \in S} \tilde{F}_k(i)} = c_k \,.\, \tilde{F}_k(j)\,, \tag{4.11}$$

where

$$\begin{aligned}
\tilde{F}_1(j) &= F_1(j)\,, \\
\tilde{F}_k(j) &= b_{j,s_k} \sum_{i \in S} F^{(s)}_{k-1}(i)\, p_{ij} \qquad \text{for each } k = 2, \ldots, n\,, \\
c_k &= \frac{1}{\sum\limits_{i \in S} \tilde{F}_k(i)} \qquad\qquad\qquad \text{for each } k = 1, \ldots, n\,.
\end{aligned}$$

Next the backward equations need to be scaled. These are scaled for each time point $k = 1, \ldots, n$ using the *same* scaling parameters which were used for the forward equations. That is define, for each $i \in S$ and $k = 1, \ldots, n$

$$B^{(s)}_k(i) = c_k \,.\, \tilde{B}_k(i)\,, \tag{4.12}$$

where

c_k was defined in (4.11),

$\tilde{B}_n(i) = B_n(i) = 1$,

$\tilde{B}_k(i) = \sum\limits_{j \in S} b_{j,s_{k+1}}\, B^{(s)}_{k+1}(j)\, p_{ij} \qquad \text{for each } k = 1, \ldots, n-1\,.$

By repeatedly making use of equations (4.11) and (4.12), it can easily be proven through induction that, for each $k = 1, \ldots, n$ and $i \in S$,

$$\begin{aligned}
F^{(s)}_k(i) &= \left(\prod_{t=1}^{k} c_t \right) F_k(i) = C_k F_k(i)\,, \text{ and} \\
B^{(s)}_k(i) &= \left(\prod_{t=k}^{n} c_t \right) B_k(i) = D_k B_k(i)\,.
\end{aligned} \tag{4.13}$$

Since each scaling factor effectively restores the magnitude of the forward equations to 1, and since the magnitudes of the forward and backward equations are comparable (the same scaling factors were used for both the forward and backward equations), the above scaling procedure is an effective way of keeping the computations within

reasonable bounds. Furthermore, the scaling procedure described above will ensure that the re-estimation equations of the BWA are unaffected.

To see this, notice that the $\xi_k(i,j)$ and $\gamma_k(i)$ terms form the building blocks of the BWA re-estimation equations. Hence, if it can be shown that the described scaling procedure does not influence $\xi_k(i,j)$ and $\gamma_k(i)$ for each $k = 1, \ldots, n$ and $i, j \in S$, it will imply that the scaling procedure will not influence the results of the BWA. To this end, using the scaled forward and backward equations to calculate $\xi_k(i,j)$ (denoted $\xi_k^{(s)}(i,j)$), the following is obtained:

$$\begin{aligned}
\xi_k^{(s)}(i,j) &= \frac{F_k^{(s)}(i)\, p_{ij}\, b_{j,s_{k+1}}\, B_{k+1}^{(s)}(j)}{\sum_{l \in S} F_n^{(s)}(l)} \quad \text{.....} \quad \text{by (3.2) and (4.4)} \\
&= \frac{C_k\, F_k(i)\, p_{ij}\, b_{j,s_{k+1}}\, D_{k+1}\, B_{k+1}(j)}{\sum_{l \in S} C_n\, F_n(l)} \quad \text{.....} \quad \text{by (4.13)} \\
&= \frac{C_n\, F_k(i)\, p_{ij}\, b_{j,s_{k+1}}\, B_{k+1}(j)}{C_n \sum_{l \in S} F_n(l)} \\
&= \frac{F_k(i)\, p_{ij}\, b_{j,s_{k+1}}\, B_{k+1}(j)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \quad \text{.....} \quad \text{by (3.2)} \\
&= \xi_k(i,j) \quad \text{.....} \quad \text{by (4.4).}
\end{aligned}$$

Similarly, using the scaled forward and backward equations to calculate $\gamma_k(i)$ (denoted $\gamma_k^{(s)}(i)$), the following is obtained:

$$\begin{aligned}
\gamma_k^{(s)}(i) &= \frac{F_k^{(s)}(i) B_k^{(s)}(i)}{\sum_{l \in S} F_k^{(s)}(l) B_k^{(s)}(l)} \quad \text{.....} \quad \text{by (3.5) and (4.4)} \\
&= \frac{C_k\, D_k\, F_k(i)\, B_k(i)}{C_k\, D_k \sum_{l \in S} F_k(l) B_k(l)} \quad \text{.....} \quad \text{by (4.13)} \\
&= \frac{F_k(i)\, B_k(i)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \quad \text{.....} \quad \text{by (3.5)} \\
&= \gamma_k(i) \quad \text{.....} \quad \text{by (4.4).}
\end{aligned}$$

And so it is proven that the scaling procedure described in this section will not influence the HMM parameter estimates calculated by the BWA.

4.1.2.2 Initialising parameters in the BWA

It has been previously mentioned that the estimates computed by the BWA will produce a local maxima of the likelihood function. As such, the initial estimates chosen for the BWA will influence the final calculated Baum-Welch estimates. The key question then is whether the initial parameters can be chosen such that the local maxima found by BWA is equal to (or close to) the global maxima of the likelihood function. Unfortunately however, at the time of writing, no solution to this question could be found in the literature.

One option available to overcome this is to choose (randomly or otherwise) several different initial parameter sets. The BWA can then be performed for each initial parameter set, with the final parameter estimates being the estimates which yield the largest local maxima.

Another technique which is also discussed in the literature (see for example [37]) is segmentation of the observation sequence into states. This is achieved by firstly choosing an initial estimate for the model parameter set, denoted $\lambda^{*(1)}$ (this is usually done randomly). $\lambda^{*(1)}$ together with the observation sequence is then used to perform the BWA and produce $\hat{\lambda}^{(1)}$, the parameter set which yields a local maxima. Using the observation sequence and $\hat{\lambda}^{(1)}$, the Viterbi algorithm (see Section 3.2) is performed - thus calculating an optimal state path. This state path can then be used to segment the observation sequence into states. Thus the proportion of times a given signal was emitted from a given state can be estimated. The proportion of times a given state transition occurred can also be estimated from the Viterbi state path. Based on these, improved initial estimates (denoted $\lambda^{*(2)}$) can be determined for the BWA. The observation sequence can once again be used together with $\lambda^{*(2)}$ to perform the

BWA and produce $\hat{\lambda}^{(2)}$, the parameter set which yields the local maxima given the initial estimate $\lambda^{*(2)}$. This process can then be iteratively repeated until convergence in $\hat{\lambda}^{(\cdot)}$ is achieved. The process described above can however become quite involved since we now have two iterative processes - the iterative process of the BWA nested within the iterative process of choosing the initial parameters.

The segmentation of the observation sequence into the states can usually be done manually if the signal space is discrete. This however will not be possible if the signal space is continuous. Under such applications [37] suggests maximum likelihood segmentation or segmentation using k-means clustering to cluster the observed signals within each state.

Regardless of how the initial parameter set is chosen for the BWA, it is important that the chosen parameters satisfy the constraints given in equation (4.10). As has been previously mentioned, if the initial estimates satisfy these constraints, then these constraints will also be satisfied for each iteration of the BWA.

The effect of varying initial parameter estimates on the final BWA estimates may be of interest. This was investigated through a simulation exercise and the results are presented in Section 7.1.1 of this dissertation.

4.1.2.3 Insufficient training data

A potential challenge associated with training HMM parameters is that the observation sequence is finite. Thus there may be instances where there are insufficient occurrences of certain model events (e.g. signal occurrences within certain states) to efficiently estimate certain model parameters.

The simplest solution to this problem is to simply increase the size of the training observation set. This could include increasing the length of the training observation sequence and/or using multiple observation sequences (this is discussed in Section 4.1.3) when training the HMM. However, in many applications this may be imprac-

tical or expensive.

A second possible solution is to reduce the size of the model - e.g. reduce the number of states or the number of possible signals per state. Typically the number of parameters can also be reduced by assuming a distribution HMM (since, for a given state, the parameters for the assumed signal distribution now need to be estimated rather than the actual signal probability for each possible signal). While reducing the number of parameters is usually possible, doing so may increase the risk of model misspecification.

A consequence of insufficient training data may be that one or more parameters of the HMM are incorrectly estimated to be zero or very close to zero. To see this, consider a HMM where it is likely that signal $\nu_m \in \delta$ will be emitted if the state sequence is in state $j \in S$. However, due to insufficient training data, no such events have occurred during the time points for which the HMM has been observed, and as such b_{jm} is estimated to be zero.

Now suppose that it is required to calculate the probability that a given new signal sequence will be generated by the model. Even if this new sequence is likely to be generated by the model, due to parameters which have been estimated to be zero, the probability of this signal sequence being observed (given the estimated model parameters) may be calculated to be zero, therefore indicating an impossible event.

To illustrate this, consider the HMM example which was given in Section 2.1.1. For this HMM the actual signal matrix is given as $\mathbf{B} = \begin{pmatrix} 0.99 & 0.01 \\ 0.96 & 0.04 \end{pmatrix}$. Suppose that the signal sequence $\mathbf{s}_5 = (\nu_1, \nu_1, \nu_1, \nu_1, \nu_1)$ has been observed and is used to train the BWA. This will result in the BWA estimate $\hat{\mathbf{B}} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$. Now suppose that it is desired to calculate, for a similar machine, the probability that $\mathbf{s}_3 = (\nu_1, \nu_2, \nu_1)$ will be observed. Then using the above $\hat{\mathbf{B}}$, this probability will be calculated as $P(\mathbf{S}_3 = (\nu_1, \nu_2, \nu_1)|\hat{\lambda}) = 0$, thereby indicating an impossible event. However, in Section 3.1.3 it was shown that the true probability of this event is in fact

$P(\mathbf{S}_3 = \mathbf{s}_3|\lambda) = 0.0174$. This non-zero probability specifies that $\mathbf{s}_3 = (\nu_1, \nu_2, \nu_1)$ is in actual fact not an impossible event.

The error described could be fatal in certain applications of the HMM, yet the shorter the observation sequence used to train the HMM, the more likely it is to occur. A solution is given in [31] in which it is proposed that the additional constraint $0 < \epsilon \leq b_{jk} \leq 1$ be applied to the estimated HMM parameters for each $j \in S$ and $\nu_k \in \delta$. To incorporate this constraint, [31] proposes that the Baum-Welch re-estimation equations ((4.7) to (4.9)) be used to estimate $\mathbf{B}$, denoted by $\hat{\mathbf{B}}$. Now suppose that there are N signals in the signal space δ and that $l < N$ of the estimated parameters of the j^{th} row of $\hat{\mathbf{B}}$ are less than ϵ. It can be assumed without loss of generality that these correspond to the first l signals in δ. So $\hat{b}_{jk} < \epsilon$ for $1 \leq \nu_k \leq l$. Now set $\acute{b}_{jk} = \epsilon$ for $1 \leq \nu_k \leq l$ and re-align the remaining parameters in the j^{th} row so that they sum to $(1 - l\epsilon)$. This can be achieved as follows

$$\acute{b}_{jk} = (1 - l\epsilon)\frac{\hat{b}_{jk}}{\sum_{i=l+1}^{N} \hat{b}_{ji}} \qquad \text{for } \nu_k = l+1, \ldots, N\,.$$

If one or more $\acute{b}_{jk}$ become less than ϵ (for $\nu_k = l+1, \ldots, N$) when the above re-alignment is performed, then these values must also be set equal to ϵ and the remaining $\acute{b}_{jk}$ re-aligned.

After performing the above re-alignment for each row of $\hat{\mathbf{B}}$, it can easily be verified that the resulting $\acute{\mathbf{B}}$ will satisfy the constraints $0 < \epsilon \leq \acute{b}_{jk} \leq 1$ and $\sum_{i=l}^{N} \acute{b}_{ji} = 1$ for each $j \in S$ and $\nu_k \in \delta$. Furthermore it is also verified in [31] that $\acute{\mathbf{B}}$ is the value for $\mathbf{B}$ which will maximise the likelihood $P(\mathbf{S}_n = \mathbf{s}_n \,|\, \lambda)$ subject to the desired constraints.

If required, this methodology can similarly be extended to also include the other parameters of λ.

4.1.3 The Baum-Welch Algorithm for Multiple Observation Sequences

In certain applications of the HMM, multiple observation sequences (all of which were generated by the same HMM) may be available to train the unknown parameter set λ. An example of such an application is speech recognition, of which the voice dialling feature in almost all modern cellular phones is a fairly common and well-known example. For voice dialling, a separate HMM is built for each word in the vocabulary. When training the HMM for a given word, the word is spoken several times by the user of the cellular phone. Each time it is spoken the word is converted into a observation or signal sequence. In this way multiple observation sequences are available to train the HMM for that particular word.

Regardless of what the application of the HMM may be, should multiple observation sequences be available, it would be desirable that all the available data be used to estimate the parameters of the HMM. As such, this section will discuss how the BWA can be adapted to train the HMM when multiple observation sequences, all of which were generated from the HMM in question, are available.

To begin, assume that M observation sequences have been generated by the same HMM and that the set of these observation sequences is notated by

$$\acute{\mathbf{S}} = [\mathbf{S}_{n_1}^{(1)}, \mathbf{S}_{n_2}^{(2)}, ..., \mathbf{S}_{n_M}^{(M)}],$$

where $\mathbf{S}_{n_r}^{(r)} = (s_1^{(r)}, s_2^{(r)}, ..., s_{n_r}^{(r)})$ is the r^{th} observation sequence, consisting of n_r individual signals (observations), and $r \in \{1, 2, ..., M\}$.

It is assumed that each observation sequence was generated independently of every other observation sequence. The goal of the BWA now becomes to find the HMM that has the highest likelihood of generating all M observation sequences, that is to estimate the parameters of λ such that the following likelihood is maximised:

$$P(\acute{\mathbf{S}} \,|\, \lambda) \;=\; \prod_{r=1}^{M} P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \lambda)\,. \tag{4.14}$$

As was the case for a single observation sequence, define $\lambda^* = (\mathbf{P}^*, \mathbf{B}^*, \mathbf{a}^*)$ to be the current estimate of the parameter set for the HMM, and $\hat{\lambda} = (\hat{\mathbf{P}}, \hat{\mathbf{B}}, \hat{\mathbf{a}})$ to be the Baum-Welch re-estimate of this parameter set.

Also, let

$$\xi_k^{*(r)}(i,j),\ \gamma_k^{*(r)}(i),\ \gamma_{k,m}^{*(r)}(i)$$

be the probabilities corresponding to those given in equation (4.3), calculated for the r^{th} observation sequence using λ^*, where $r = 1, 2, ..., M;\ k = 1, ..., n_r;\ i, j \in S$; and $\nu_m \in \delta$.

Since the Baum-Welch re-estimation equations for a single observation sequence are based on the expected number of occurrences of certain events, it is suggested in [31] that the re-estimation equations for multiple observation sequences be modified by adding together the individual frequencies of these occurrences for each of the M sequences. The modified re-estimation formulas for multiple observation sequences are thus

$$\hat{p}_{ij} = \frac{\sum_{r=1}^{M}\sum_{k=1}^{n_r-1}\xi_k^{*(r)}(i,j)}{\sum_{r=1}^{M}\sum_{k=1}^{n_r-1}\gamma_k^{*(r)}(i)} \qquad \text{for each } i,j \in S$$

$$\hat{b}_{jm} = \frac{\sum_{r=1}^{M}\sum_{k=1}^{n_r}\gamma_{k,m}^{*(r)}(j)}{\sum_{r=1}^{M}\sum_{k=1}^{n_r}\gamma_k^{*(r)}(j)} \qquad \text{for each } j \in S \text{ and } \nu_m \in \delta. \tag{4.15}$$

It is mentioned in [37] that the BWA estimates in equation (4.15) will locally maximise the likelihood function expressed in equation (4.14).

In the case of a single observation sequence, when $\hat{p}_{ij}$ and $\hat{b}_{jm}$ is calculated, the term $1/P(\mathbf{S}_n = \mathbf{s}_n|\lambda^*)$ appears in both the numerator and the denominator and can therefore be cancelled out. This however is not the case for multiple observation

sequences, as is shown for $\hat{p}_{ij}$ below:

$$\hat{p}_{ij} = \frac{\sum_{r=1}^{M}\sum_{k=1}^{n_r-1}\xi_k^{*(r)}(i,j)}{\sum_{r=1}^{M}\sum_{k=1}^{n_r-1}\gamma_k^{*(r)}(i)} = \frac{\sum_{r=1}^{M}\frac{1}{P^{*(r)}}\sum_{k=1}^{n_r-1}F_k^{*(r)}(i)\,p_{ij}^{*}\,b^{*}_{j,s_{k+1}^{(r)}}\,B_{k+1}^{*(r)}(j)}{\sum_{r=1}^{M}\frac{1}{P^{*(r)}}\sum_{k=1}^{n_r-1}F_k^{*(r)}(i)B_k^{*(r)}(i)}$$

where $P^{*(r)} = P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \mid \lambda^*)$ for $r = 1, 2, \ldots, M$.

It should be clear that the re-estimation equations in (4.15) will satisfy $\hat{p}_{ij} \geq 0$ and $\hat{b}_{jm} \geq 0$. It can also be proven algebraically that $\sum_{j\in S}\hat{p}_{ij} = 1$ and $\sum_{\nu_m\in\delta}\hat{b}_{jm} = 1$ since $\sum_{j\in S}\xi_k^{*(r)}(i,j) = \gamma_k^{*(r)}(i)$ and $\sum_{\nu_m\in\delta}\gamma_{k,m}^{*(r)}(j) = \gamma_k^{*(r)}(j)$ for a given r and k.

Should scaling be required, the scaling technique previously detailed in Section 4.1.2 can once again be used to scale the forward and backward equations. Since the $1/P^{*(r)}$ terms are left in the re-estimation equations, the scaling factors will be canceled for each term within the inner summation (see Section 4.1.2 for a proof of this). Thus using scaled forward and backward equations when computing the re-estimation equations will correctly result in unscaled $\hat{p}_{ij}$ and $\hat{b}_{jm}$.

In [31], from which the re-estimation equations in (4.15) are adapted, the application of the HMM was such that it was convenient to assume that the initial state was always state 1; that is that $p_1 = 1$ and $p_i = 0$ for all $i \neq 1$. Therefore a re-estimation equation for p_i was not included in [31]. To align with the re-estimation equations in (4.15), this dissertation suggests the following re-estimation equation for p_i:

$$\begin{aligned}\hat{p}_i &= \text{the average of } \{P(X_k = i|\mathbf{S}_{n_1}^{(1)} = \mathbf{s}_{n_1}^{(1)}, \lambda)\,,\ \ldots\ ,\ P(X_k = i|\mathbf{S}_{n_M}^{(M)} = \mathbf{s}_{n_M}^{(M)}, \lambda)\} \\ &= \frac{1}{M}\sum_{r=1}^{M}\gamma_1^{*(r)}(i)\,.\end{aligned}$$

Using this re-estimation equation will also ensure that $\hat{p}_i \geq 0$ and $\sum_{i\in S}\hat{p}_i = 1$ for each iteration. This estimate can also be enhanced by incorporating a weighting for each

$\gamma_1^{*(r)}(i)$, for example a weighting based on the sequence length of $\mathbf{S}_{n_r}^{(r)}$ or the likelihood value $P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \lambda)$.

This is similar to what is suggested in [18] where an alternative approach of using multiple observation sequences to train the BWA is given. In [18] it is suggested that the parameters of λ first be estimated using the single sequence Baum-Welch re-estimation equations (equations (4.7) to (4.9)) for each individual observation sequence. Thus M different estimates are obtained for each parameter. The final Baum-Welch estimates are then given by:

$$\begin{aligned}
\hat{p}_i &= \sum_{r=1}^{M} \frac{W_r}{N_a}\, \hat{p}_i^{(r)} \\
\hat{p}_{ij} &= \sum_{r=1}^{M} \frac{W_r}{N_b}\, \hat{p}_{ij}^{(r)} \\
\hat{b}_{jm} &= \sum_{r=1}^{M} \frac{W_r}{N_c}\, \hat{b}_{jm}^{(r)}\,,
\end{aligned}$$

where $\hat{\lambda}^{(r)} = \left(\hat{\mathbf{P}}^{(r)}, \hat{\mathbf{B}}^{(r)}, \hat{\mathbf{a}}^{(r)}\right)$ is the final Baum-Welch estimate obtained from $\mathbf{S}_{n_r}^{(r)}$,
W_r is the weighting factor for the estimates from $\mathbf{S}_{n_r}^{(r)}$,
N_a, N_b and N_c are normalization factors.

The effectiveness of several different weightings was tested in [18]. These included unit weight factors ($W_r = 1$ for each observation sequence), weight factors expressed as a function of $P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \hat{\lambda}^{(r)})$ and weight factors expressed as a function of $P(\acute{\mathbf{S}} \,|\, \hat{\lambda}^{(r)})$. For each of these weightings, 'trimmed' weight factors were also tested whereby the weight factors for unlikely models were set to 0. That is, for each $r \in \{1, 2, \ldots, M\}$, either $P(\acute{\mathbf{S}} \,|\, \hat{\lambda}^{(r)})$ or $P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \hat{\lambda}^{(r)})$ was calculated and the models were ranked accordingly. For the lowest ranked models, $W_r = 0$ was used. In this way, poorly estimated HMMs (on a sequence-by-sequence basis) were eliminated from the final parameter estimation.

In particular, the performance of the re-estimation equations using nine different weight factors was tested against the re-estimation equations given in [31] (equation (4.15)). According to the results documented in [18], two weight factors produced model estimates which out-performed the model estimates obtained using the re-estimation equations given in [31] (the performance measure used in [18] was the value of the likelihood function in equation (4.14), evaluated using the estimated model parameters). These weight factors are the unit weight factors ($W_r = 1$ for each r) and one of the 'trimmed' unit weight factors ($W_r = 1$ for the higher ranked models, otherwise $W_r = 0$).

These results may however not necessarily hold for all HMMs. It is therefore advised that several of the above mentioned techniques from [18] and [31] be considered when estimating the HMM parameters using multiple observation sequences.

Finally, the estimation methods discussed in this section have assumed that the multiple observation sequences were generated independently of each other; [32] presents an approach for training HMMs using multiple observation sequences without imposing this assumption.

4.1.4 The Baum-Welch Algorithm for Distribution Hidden Markov Models

In Section 2.2 distribution HMMs were described. Recall that these HMMs have the same properties and assumptions as the general HMM; the only difference being that given the state at a particular point in time, it is assumed that the observation is emitted according to a probability distribution - e.g. a binomial or a Poisson distribution. For example the Poisson HMM assumes that given the HMM is in state j at some time point t, the probability of observing the signal $x \in \{0, 1, 2, \ldots\}$ is defined as

$$P[S_t = x | X_t = j] = b_{jx} = \frac{e^{-\omega_j}\, \omega_j^x}{x!} \tag{4.16}$$

for each $j \in S$. This probability was also detailed in equation (2.8) of Section 2.2.

Since the state process for a distribution HMM is assumed to have the same properties as the general HMM, and the state process of a HMM is in no way influenced by the observed signals, the Baum-Welch equations for p_i and p_{ij} (previously defined in equations (4.7) and (4.8)) are once again the Baum-Welch equations for the distribution HMM. This can also be seen through the mathematical derivation of the BWA equations shown in Appendix B.

The Baum-Welch equation for b_{jm} for the general HMM, defined in equation (4.9), will however no longer be appropriate for the distribution HMM. For the general HMM, the focus of the BWA is to estimate b_{jm} individually for each $j \in S$ and $\nu_m \in \delta$. For the distribution HMM, the focus now shifts to finding the appropriate distribution parameters for each $j \in S$ (e.g. ω_j for each $j \in S$ for the Poisson HMM). For a given state j, once the appropriate distribution parameters have been estimated, these can be used to estimate b_{jx} for $x \in \delta$.

It is shown for the general HMM in Appendix B that the BWA estimate (for a given iteration of the algorithm) for b_{jm} is derived by maximising

$$\sum_{k \in S} \sum_{t=1}^{n} \ln(b_{k,s_t})\, P(X_t = k | \, \mathbf{S}_n = \mathbf{s}_n, \lambda^*) \tag{4.17}$$

with respect to b_{jm}, subject to the constraint $\sum\limits_{\nu_k \in \delta} b_{jk} = 1$. This is achieved through differentiation with respect to b_{jm} . It is shown in Appendix B that iteratively estimating b_{jm} in this way will result in the local maximization of the likelihood $P(\mathbf{S}_n = \mathbf{s}_n \,|\, \lambda)$.

For the distribution HMM, b_{k,s_t} in equation (4.17) is replaced by the appropriate probability distribution (e.g. equation (4.16) for the Poisson HMM) and maximisation is now performed with respect to the appropriate distribution parameters (e.g. ω_j for the Poisson HMM), subject to the required constraints of the distribution parameters (e.g. $\omega_j \geq 0$ for the Poisson HMM). This is performed for each $j \in S$. The constraint $\sum\limits_{x \in \delta} b_{jx} = 1$ will be implicitly satisfied for each $j \in S$ as this is a property of all valid

probability distributions.

The Baum-Welch estimation of b_{jx} for the Poisson HMM, for $j \in S$ and $x \in \{0, 1, 2, \ldots\}$, is now derived.

For the Poisson HMM it can easily be verified that equation (4.17) can be simplified, through the use of equation (4.16), to the following:

$$\begin{aligned} & \sum_{k \in S} \sum_{t=1}^{n} \ln\left(\frac{e^{-\omega_k}\, \omega_k^{s_t}}{s_t!}\right) P(X_t = k | \, \mathbf{S}_n = \mathbf{s}_n, \lambda^*) \\ = & \sum_{k \in S} \sum_{t=1}^{n} \{(-\omega_k + s_t \ln(\omega_k) - \ln(s_t!)) \; P(X_t = k | \, \mathbf{S}_n = \mathbf{s}_n, \lambda^*)\} \; . \end{aligned} \tag{4.18}$$

For a given state $j \in S$, the maximisation of (4.18) with respect to ω_j can be achieved through differentiation as follows:

$$\begin{aligned} \frac{\partial}{\partial \omega_j}\left[\sum_{k \in S} \sum_{t=1}^{n} \ln\left(\frac{e^{-\omega_k}\, \omega_k^{s_t}}{s_t!}\right) P(X_t = k | \, \mathbf{S}_n = \mathbf{s}_n, \lambda^*)\right] &= 0 \\ \frac{\partial}{\partial \omega_j}\left[\sum_{k \in S} \sum_{t=1}^{n} \{(-\omega_k + s_t \ln(\omega_k) - \ln(s_t!)) \; \gamma_t^*(k)\}\right] &= 0 \quad \ldots\ldots \text{ by (4.3), (4.6) and (4.18)} \\ \frac{\partial}{\partial \omega_j}\left[\sum_{t=1}^{n} \{(-\omega_j + s_t \ln(\omega_j) - \ln(s_t!)) \; \gamma_t^*(j)\}\right] + 0 &= 0 \\ \sum_{t=1}^{n} \left\{-\gamma_t^*(j) + \frac{s_t}{\omega_j}\, \gamma_t^*(j)\right\} &= 0 \\ \Rightarrow \hat{\omega}_j &= \frac{\sum_{t=1}^{n} \gamma_t^*(j)\, s_t}{\sum_{t=1}^{n} \gamma_t^*(j)} \; . \end{aligned}$$

The above assumes that $\sum_{t=1}^{n} \gamma_t^*(j) \neq 0$. As $\gamma_t^*(j) \geq 0$ for each time point t, this will hold if $\gamma_t^*(j) > 0$ for at least one t. Of course $\gamma_t^*(j) = 0$ for each t, implies that $P(X_t = j | \, \mathbf{S}_n = \mathbf{s}_n, \lambda^*) = 0$ for each t, which then questions either the validity of the estimate λ^* (perhaps improved initial estimates should be chosen) or the validity of keeping state j in the model. Either way the assumption that $\sum_{t=1}^{n} \gamma_t^*(j) \neq 0$ seems

reasonable for a well functioning Poisson HMM.

Also note that $\hat{\omega}_j \geq 0$ is satisfied, a necessary constraint for the Poisson distribution.

To show that $\hat{\omega}_j$ is indeed a maxima, the second partial derivative is evaluated at $\hat{\omega}_j$. This yields

$$\begin{aligned}
& \frac{\partial^2}{\partial {\omega_j}^2} \left[\sum_{k \in S} \sum_{t=1}^{n} \ln \left(\frac{e^{-\omega_k}\, \omega_k^{s_t}}{s_t!} \right) P(X_t = k | \, \mathbf{S}_n = \mathbf{s}_n, \lambda^*) \right] \\
= & \frac{\partial}{\partial \omega_j} \left[\sum_{t=1}^{n} \left\{ -\gamma_t^*(j) + \frac{s_t}{\omega_j}\, \gamma_t^*(j) \right\} \right] \\
= & -\frac{1}{{\omega_j}^2} \sum_{t=1}^{n} \gamma_t^*(j)\, s_t \,.
\end{aligned}$$

And so

$$\begin{aligned}
& \frac{\partial^2}{\partial {\omega_j}^2} \left[\sum_{k \in S} \sum_{t=1}^{n} \ln \left(\frac{e^{-\omega_k}\, \omega_k^{s_t}}{s_t!} \right) P(X_t = k | \, \mathbf{S}_n = \mathbf{s}_n, \lambda^*) \right] \Bigg|_{\omega_j = \hat{\omega}_j} \\
= & -\frac{\left[\sum_{t=1}^{n} \gamma_t^*(j) \right]^2}{\sum_{t=1}^{n} \gamma_t^*(j)\, s_t} \\
< & \; 0 \text{ if } \gamma_t^*(j)\, s_t \neq 0 \text{ for at least one } t = 1, 2, \ldots, n \,.
\end{aligned}$$

If $\gamma_t^*(j)\, s_t = 0$ for each $t = 1, 2, \ldots, n$ then inspection of equation (4.18) reveals that the value for ω_j (subject to $\omega_j \geq 0$) which will result in the maximisation of (4.18) is $\omega_j = 0 = \hat{\omega}_j$. And so the above derived $\hat{\omega}_j$ is indeed a maxima.

Once $\hat{\omega}_j$ has been calculated for each $j \in S$, equation (4.16) can be used to calculate $\hat{b}_{jx}$ for each $j \in S$ and $x \in \{0, 1, 2, \ldots\}$. Therefore $\hat{b}_{j,s_k}$ can be calculated for each $j \in S$ and each $k = 1, 2, \ldots, n$. By making use of the forward and backward equations, this in turn can be used to calculate the final value of the likelihood, $P(\mathbf{S}_n = \mathbf{s}_n \,|\, \hat{\lambda})$, for the iteration of the BWA in question (see Section 3.1).

For the normal HMM, given that the HMM is in state $j \in S$ at some time point t,

the probability of observing the signal $x \in \mathbb{R}$ is defined as

$$P[S_t = x | X_t = j] = b_{jx} = \left(2\pi\sigma_j^2\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma_j^2}(x-\mu_j)^2\right\}.$$

Using similar techniques to those used for the Poisson HMM, differentiation of (4.17) with respect to μ_j and σ_j^2 can be performed to yield the maximizing values of μ_j and σ_j^2 :

$$\begin{aligned}
\hat{\mu}_j &= \frac{\sum_{t=1}^{n} \gamma_t^*(j)\, s_t}{\sum_{t=1}^{n} \gamma_t^*(j)}, \text{ and} \\
\hat{\sigma}_j^2 &= \frac{\sum_{t=1}^{n} \gamma_t^*(j)\, (s_t - \hat{\mu}_j)^2}{\sum_{t=1}^{n} \gamma_t^*(j)}.
\end{aligned}$$

This holds true for each $j \in S$.

Further variations of distribution HMMs are also discussed in [35] and [46]. In particular [46] (see pages 116 to 118) discusses when different state-dependent distributions (i.e. the distributions which are assumed to emit the signals) are appropriate. It is suggested that Poisson and negative binomial HMMs be considered when the observed signals are unbounded counts, Bernoulli HMMs be considered for binary observations and binomial HMMs be considered when the observed signals are bounded counts. In this discussion, it is also noted that exponential, normal and Gamma distributions are important state-dependent distributions for the HMM when continuous-valued signals are observed. For example, in Section 13.2 of [46] a normal HMM is used to model share return series for four shares.

Finally it is noted in [46] (see page 66) that the ease by which equation (4.17) can be maximised depends on the state-dependent distribution assumed. For example, in the case of the Poisson and normal distributions, closed-form solutions are available (as has been demonstrated in this section). In other cases, e.g. gamma and negative

binomial distributions, numerical techniques are required to carry out the necessary maximisation.

4.2 Solving the Learning Problem Through Direct Maximization of the Likelihood

Section 4.1 of this dissertation discussed using the BWA as a tool to solve the learning problem, thereby estimating the model parameters of a HMM. This section (based primarily on [34] and [46]) will summarise an alternative methodology, that of using numerical techniques to directly estimate the parameter set λ which will maximise the likelihood $P(\mathbf{S}_n = \mathbf{s}_n \,|\, \lambda)$.

In a general maximisation framework, several different numerical techniques exist which can be called upon to maximise the likelihood. These are implemented in various software packages. For example the unconstrained optimisers `nlm` and `optim` are available in **R** as well as the package `constrOptim` which permits constraints to be placed on the parameters which need to be optimised. For a given application of the HMM it is advised that several different techniques be explored.

In many applications of the HMM certain considerations may need to be taken into account, whether estimating the model parameters through direct maximisation or using the BWA. These include numerical underflow in the calculation of the likelihood, constraints on the parameters which need to be estimated, and multiple local maxima in the likelihood function. These were addressed for the BWA in Section 4.1 and are discussed next for the direct maximisation approach.

To begin, recall from Section 3.1 that an effective way to calculate the likelihood function is through the use of the forward equations as follows

$$P(\mathbf{S}_n = \mathbf{s}_n|\lambda) = \sum_{i \in S} F_n(i)\,.$$

As was discussed in Section 4.1.2, numerical underflow can occur during the calculation of the forward equations. This may occur due to the fact that as k becomes large, $F_k(j)$ tends rapidly to zero. Thus, depending on the software used, the computations of the forward equations may exceed the precision range of the software for a long observed signal sequence. This may be resolved by once again making use of the scaling technique described in Section 4.1.2. To this end let L_n denote the value of the likelihood calculated using the forward equations and $L_n^{(s)}$ the value of the likelihood calculated using the scaled forward equations. Then it is shown in Section 4.1.2 that $L_n = L_n^{(s)} / C_n$, where C_n is computed during the scaling process (see equations (4.11) and (4.13)). And so, by using appropriate scaling when calculating the forward equations, the likelihood can still be computed, and therefore maximised, even if numerical underflow occurs in the calculation of the unscaled forward equations.

When maximising the likelihood function consideration of parameter constraints needs to be taken into account. Recall that various constraints are assumed for the HMM, as was discussed in Chapter 2. As mentioned, some maximisation packages can accommodate for constraints on parameters. If however unconstrained optimisers are used, re-parametrisation may be needed to guarantee that the parameter constraints are satisfied, thereby ensuring that the final HMM parameter estimates sensible. For example, each row of the state transition probability matrix must sum to one and all the state transition probabilities p_{ij} must be non-negative. To this end, a possible transformation (given in [46]) to obtain the constrained probabilities p_{ij} from re-parameterised real numbers τ_{ij}, which are unconstrained, is described below.

Let $g : \mathbb{R} \to \mathbb{R}^+$ be a strictly increasing non-negative function, e.g.

$$g(x) = e^x \quad x \in \mathbb{R}\,.$$

Now define

$$\varrho_{ij} = \begin{cases} g(\tau_{ij}), & i \neq j; \\ 1, & i = j. \end{cases}$$

By setting $p_{ij} = \varrho_{ij} / \sum_{k \in S} \varrho_{ik}$ (for each $i, j \in S$), it can easily be verified that the constraints of the transition probability matrix $\mathbf{P}$ are satisfied.

As an illustration of this, consider the first row of $\mathbf{P}$ for a three state HMM. Then,

$$\begin{aligned} p_{11} &= 1 \,/\, (1 + \exp(\tau_{12}) + \exp(\tau_{13})) \\ p_{12} &= \exp(\tau_{12}) \,/\, (1 + \exp(\tau_{12}) + \exp(\tau_{13})) \\ p_{13} &= \exp(\tau_{13}) \,/\, (1 + \exp(\tau_{12}) + \exp(\tau_{13})). \end{aligned}$$

The transformation in the opposite direction yields

$$\begin{aligned} \tau_{12} &= \ln(p_{12} \,/\, (1 - p_{12} - p_{13})) = \ln(p_{12} \,/\, p_{11}) \\ \tau_{13} &= \ln(p_{13} \,/\, (1 - p_{12} - p_{13})) = \ln(p_{13} \,/\, p_{11}). \end{aligned}$$

No transformation is required for τ_{11}, as will be explained later.

To further elaborate, let the term 'natural parameters' refer to the constrained parameters (p_{ij} in the above example) and the term 'working parameters' refer to unconstrained parameters (τ_{ij} in the above example). Then the maximisation of the likelihood can performed as follows:

- Choose initial natural parameters subject to the required constraints.
- Transform the initial natural parameters into the corresponding working parameters.
- Perform the numerical maximisation of the likelihood $P(\mathbf{S}_n = \mathbf{s}_n \,|\, \lambda)$ with respect to the working parameters.
- Transform the final working parameter estimates to the natural parameters, thereby ensuring that the final parameter estimates of the HMM satisfy the necessary constraints.

Returning to the above example, it should now be clear that the working parameter τ_{11} need not be estimated as τ_{11} is not required for the transformation back to the natural parameters.

A further comment regarding the above example is that in many applications defining $g(x)$ as

$$g(x) = \begin{cases} e^x, & x \leq 0 \\ x + 1, & x > 0 \end{cases}$$

may produce parameter estimates which are more stable than if $g(x) = e^x$ is used. The reason for this is that, due to the nature of the exponential function, defining $g(x) = e^x$ might result in small changes in the estimated working parameters (when the working parameters are greater than zero) leading to more significant changes in the estimated natural parameters, and hence possible instability in the parameter estimation process.

For the general HMM, similar transformations can also be applied to the signal probabilities b_{jm} to ensure that the final estimated probabilities satisfy the required constraints. For distribution HMMs, appropriate transformations should be applied to ensure that the parameters of the state-dependant signal distributions satisfy the necessary constraints. For example, in the case of the Poisson HMM an additional constraint, apart from the usual constraints on $\mathbf{P}$, is that the means ω_i of the state-dependant signal distributions are non-negative for each $i \in S$. One way this can be achieved is by defining the working parameters as $\eta_i = \ln(\omega_i)$ for each $i \in S$. Once the likelihood has been maximised with respect to the working parameters ($\eta_i \in \mathbb{R}$), the natural parameters can be obtained by transforming back: $\hat{\omega}_i = \exp(\hat{\eta}_i)$. And so $\hat{\omega}_i$ satisfies the required constraint.

Next the matter of multiple local maxima of the likelihood is briefly discussed. As was mentioned during the discussion of the BWA, the likelihood of the HMM is an involved function of the model parameters which will typically have several local maxima. The global maxima is of course desired. Unfortunately there is however no simple method

of guaranteeing that a given numerical maximisation approach will find the global maxima. The reason for this is that, depending on the initial parameter values, a given numerical algorithm will typically identify some local maxima rather than the desired global maxima. As mentioned in Section 4.1.2, this also applies when the BWA is used to estimate the model parameters. A sensible strategy to overcome this is to consider a range of starting values for the numerical maximisation and analyze the resulting likelihood maxima and parameter estimates.

4.3 Further Discussions Around the Learning Problem

4.3.1 Comparison of the Baum-Welch and Direct Maximization Methods

The previous two sections have described two different techniques of estimating MLEs for the HMM parameters; that being the BWA described in Section 4.1 (where an analytical approach derived from the EM algorithm is used) and an approach which considers direct numerical maximisation of the likelihood function, as described in Section 4.2. Regarding these two approaches, [34] notes that "There is a close historical connection between hidden Markov (chain) models and the EM algorithm, as the Baum-Welch algorithm for finding MLEs in such a model is an important forerunner and special case of EM". Following this, [34] further notes that "the likelihood is easy to evaluate, and although direct numerical maximisation seems less common than EM, it has by now been used fairly widely in the fitting of HMMs and extensions thereof". Interestingly [15] notes that "although neither algorithm is superior to the other in all respects, researchers and practitioners who work with HMMs tend to use only one of the two algorithms, and ignore the other".

When it is the general HMM which is being considered, the BWA has the advantage that no evaluation or maximisation of the likelihood needs to be performed directly;

that is the established BWA equations ((4.7)-(4.9)) are used to estimate the model parameters. However, as was highlighted in Section 4.1.1 (and is further discussed in Appendix B), if the HMM is assumed stationary then the BWA equations for the $\hat{p}_i$ and $\hat{p}_{ij}$ parameters (equations (4.7) to (4.8)) will no longer hold. Typically numerical techniques will have to be incorporated into the BWA under these conditions (see Appendix B for more details). To this end [34] notes that, under the assumption of stationarity, direct numerical maximisation provides a less complicated approach to parameter estimation.

When distribution HMMs are being considered, the BWA equations for the parameters of the state-dependant signal distributions will differ depending on the distribution which is assumed (see Section 4.1.4 for more details). For a given signal distribution, differentiation first needs to be performed to establish the BWA equations for that specific distribution. As was discussed in Section 4.1.4, depending on the signal distribution chosen, this may or may not prove challenging. For some signal distributions (e.g. gamma and negative binomial) no closed-formed solutions exist for the necessary differentiation, and hence numerical techniques are required to evaluate these derivatives. As such, significant changes in code may be required if it is desired that different signal distributions are tested to determine which distribution best describes an observed signal sequence and ultimately gives rise to the most meaningful parameter estimates. This however will not be the case if direct numerical maximisation of the likelihood is used to estimate the model parameters, as no differentiation is required. Hence one can repeatedly modify a model in an interactive search for the most appropriate signal distribution. Often all that is required is a change to the code which evaluates the likelihood.

However, if direct maximisation is used, one does need to take into consideration the impact that the choice of i) re-parameterisation (to ensure parameter constraints are met) and ii) numerical techniques used can have on the final estimates. This may result in a significant amount of testing.

Further discussions on this topic are provided in [15], [34] and [46]. In particular, [15] uses both simulated and actual data to investigate and compare the speed of convergence, stability, dependence on initial values, the effects of different parameterisations and the general performance for these two approaches. A hybrid algorithm combining these two approaches is also considered. As an alternative to the two approaches discussed in this chapter, Bayesian estimation is also considered and discussed in [46].

4.3.2 Standard Errors and Confidence Intervals for the Estimated Parameters

Sections 4.1 and 4.2 of this dissertation provided a discussion on how point estimators for the HMM parameters can be computed. Standard errors and intervals for these estimates may also be desirable in applications of the HMM. A discussion is provided in [46] (see Section 3.6) as to how these standard errors and intervals can be estimated, either through the use of the Hessian matrix or through the use of bootstrapping techniques. This discussion is summarised below.

Conditions under which the MLEs of the HMM parameters can be assumed asymptotically normal are discussed in [46]. If asymptotic normality can indeed be assumed, and if the standard errors of the MLEs can be estimated, then approximate confidence intervals can be computed. A suggestion in [46] is that the standard errors be estimated through the Hessian of minus the log-likelihood, evaluated at the minimum of this function, i.e. the observed information matrix. This can be outputted by many statistical packages. The inverted Hessian provides an estimate of the asymptotic variance-covariance matrix for the estimators of the HMM parameters. It is however noted that difficulties do arise when some of the parameters are on the boundary of their parameter space, which may frequently occur when the HMM is fitted.

An alternative proposed in [15] and [46] is to make use of parametric bootstrapping techniques. An excellent reference to obtain further details on bootstrapping is [21].

Through the use of this technique, multiple random sampling can be used to estimate both the standard errors and confidence intervals (based on percentiles) for the HMM parameter estimates. While this technique may overcome the short-comings of the above-mentioned approach, it should be noted that the computations may on occasions be quite time intensive. Application of these bootstrapping techniques to HMM parameter estimates, for both simulated and actual data, is given in [15] and [46].

Chapter 5

Additional Considerations for the Hidden Markov Model

Various aspects of the HMM have been detailed in the previous chapters of this dissertation. This chapter summarises further discussions regarding HMMs which appear in the literature.

5.1 Model Selection and Inspection

An increase in the number of parameters of a given statistical model will typically improve the fit of the model. In practice however, it is usually desirable not to have too many parameters in the model as over-fitting the data may reduce the out-of-sample predictive power of the model. Additionally having too many parameters in the model may also be disadvantageous as these parameters will typically have to be estimated from the available data. Hence the improvement in fit of a model has to be traded off against this the number of parameters in the model. A criterion for model selection is therefore required. This is true also when HMMs are fit to available data. In addition to this, once a HMM is selected one would desire a manner to assess goodness-of-fit and the existence of outliers to ensure that the model is adequate. This section, based on discussions provided in [46], outlines these concepts for HMMs.

5.1.1 Model Selection

A challenge which naturally arises when fitting HMMs is that of choosing an appropriate model, e.g. selecting the number of states, or choosing between a general HMM (in which case the number of possible signals need to be chosen) or a distribution HMM (in which case the appropriate signal distributions need to be chosen). As mentioned in the introductory paragraph, a selection criteria is required which will consider both the fit of the model as well as the number of parameters which need to be estimated in the model.

In the general field of statistical modelling, well documented criterion for comparing models are, among others, the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). These are defined as follows:

$$\begin{aligned} AIC &= -2\ln(L) + 2p \\ BIC &= -2\ln(L) + p\ln(T)\,, \end{aligned}$$

where $\ln(L)$ is the log-likelihood of the fitted model, p denotes the number of parameters in the model and T is the number of observations. For both AIC and BIC the first term is a measure of the fit of the model and will decrease as the fit improves. The second term is a penalty term and will increase as the number of parameters increase. Hence for both AIC and BIC, typically the model with the lowest value is chosen and both the model fit and number of parameters is taken into account.

Also note that compared to AIC, the penalty term of BIC has more weight if $T > e^2 \approx 7.4$, which holds in most applications. Thus BIC can often favour models with fewer parameters when compared to AIC.

Crucially, for the purposes of this dissertation, since the likelihood $L = P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ can be readily calculated for the HMM using the forward and backward equations (see Section 3.1), AIC and BIC calculations can be used to compare HMMs of varying state and signal structures.

A worked example is given in [46] (see Section 6.1) whereby AIC and BIC are used to compare various HMMs and independent mixture models. This is done using the earthquake data which was discussed in Section 2.2 of this dissertation. For this particular example, both AIC and BIC finds the 3 state Poisson HMM to be superior to the independent mixture models and other HMMs fitted.

Finally AIC and BIC provides one method of selecting a model from a group of potential models. These criteria are by no means the only ones which can be used (for example [46] also discusses the comparison of autocorrelation functions between fitted HMMs).

5.1.2 Testing Model Adequacy with Pseudo-Residuals

Once a HMM has been selected by some criterion as the 'best' model, the question still remains as to whether the model is indeed adequate. To this end, tools to assess general goodness-of-fit and to identify possible outliers are desired. For instance, in the context of regression models, the role of the residuals as a tool to test model adequacy is well established. This section will introduce quantities called pseudo-residuals which can fulfil a similar role in a more general sense. These will prove useful in testing goodness-of-fit and detecting outliers for fitted HMMs. The general concept of pseudo-residuals will first be explained after which their application to HMMs will be reviewed.

To begin, consider the useful statistical result (proven in various references within the literature - see for example page 54 of [17]) which states:

> Let X be some continuous random variable with distribution function F.
> Then $U = F(X)$ is uniformly distributed on the unit interval, i.e. $U \sim \mathrm{U}(0, 1)$.

Now, the uniform pseudo-residual of an observation x_t from a continuous random variable X_t is defined, under the fitted model, as

$$u_t = P(X_t \leq x_t) = F_{X_t}(x_t)\,.$$

That is, the pseudo-residual u_t is the observation x_t transformed by the distribution function of the fitted model. If the fitted model adequately describes the observed data, then these pseudo-residuals are approximately distributed $\mathrm{U}(0,1)$. Conversely, if a histogram or quantile-quantile plot (qq-plot) of the calculated uniform pseudo-residuals casts doubt on the conclusion that they are distributed $\mathrm{U}(0,1)$, then one should suspect that the fitted model may indeed not be adequate.

While the uniform pseudo-residuals are useful in this respect, it has the drawback that outliers may be difficult to visually detect using them. This is because it is difficult to see if a value is unlikely or not; for instance a pseudo-residual of 0.999 (a potential outlier) is difficult to distinguish from a value of say 0.96.

This shortfall can be easily rectified if the following statistical result is considered.

> Let X be some continuous random variable with distribution function F, and Φ the standard normal distribution function. Then $Z = \Phi^{-1}(F(X))$ is distributed standard normal.

This follows from the previous mentioned result and is also discussed on page 55 of [17].

Now, the normal pseudo-residual of an observation x_t from a continuous random variable X_t is defined, under the fitted model, as

$$z_t = \Phi^{-1}(u_t) = \Phi^{-1}(F_{X_t}(x_t))\,.$$

If the observations were indeed generated from the fitted model, then the normal pseudo-residuals z_t would follow a standard normal distribution. One can therefore

test the adequacy of the fitted model by performing normality tests on these residuals. Since extreme observations are visually easier to identify when the observations follow a standard normal distribution (as opposed to the uniform distribution), potential outliers are easier to detect (from a visual sense) when using the normal pseudo-residuals rather than the uniform pseudo-residuals.

The above outlined theory has dealt with continuous distributions only. It is noted in [46] that in the case of discrete observations, it is usually more meaningful to no longer define the pseudo-residuals as points but rather as intervals (this allows for the pseudo-residuals to be plotted as a histogram, thereby enabling a visual inspection of the distribution of the pseudo-residuals). Thus for a discrete random variable X_t with distribution function F_{X_t}, the uniform pseudo-residual intervals are defined as

$$[u_t^-; u_t^+] = [F_{X_t}(x_t^-); F_{X_t}(x_t)]\,,$$

where x_t^- denotes the greatest possible realisation that is strictly less than x_t. Similarly, the normal pseudo-residual intervals as

$$[z_t^-; z_t^+] = [\Phi^{-1}(u_t^-); \Phi^{-1}(u_t^+)]\,.$$

To perform further residual analysis for discrete observations (e.g. qq-plot analysis), [46] also suggests the use of the so-called 'mid-pseudo-residuals', defined as

$$z_t^m = \Phi^{-1}\left(\frac{u_t^- + u_t^+}{2}\right)\,.$$

Now that the concept of pseudo-residuals has been outlined, pseudo-residuals in the context of HMMs can be discussed. In particular, [46] introduces two types, namely ordinary pseudo-residuals and forecast pseudo-residuals.

The ordinary pseudo-residuals for HMMs are calculated from the conditional distribution given all other observations. That is, for continuous signal observations (for example if a beta HMM is applied), the normal pseudo-residual is

$$z_t = \Phi^{-1}[P(S_t \leq s_t | \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)})]$$

where $\mathbf{S}_n^{(-t)} = (S_1, \ldots, S_{(t-1)}, S_{(t+1)}, \ldots, S_n)$. That is $\mathbf{S}_n^{(-t)}$ denotes $\mathbf{S}_n$ with t^{th} element dropped.

If the fitted HMM is adequate, then z_t should resemble a standard normal variable.

The intervals $[z_t^-; z_t^+]$ for HMMs with discrete observations follow similarly, as was explained above,

$$\begin{aligned} z_t^- &= \Phi^{-1}[P(S_t < s_t | \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)})] \\ z_t^+ &= \Phi^{-1}[P(S_t \leq s_t | \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)})] \,. \end{aligned}$$

In the discrete case, for $s_t = \nu_m \in \delta$, the conditional probabilities $P(S_t = s_t | \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)})$ can be calculated as follows:

$$\begin{aligned} P(S_t = \nu_m | \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)}, \lambda) &= \frac{P(S_t = \nu_m, \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)} | \lambda)}{P(\mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)} | \lambda)} \quad \text{..... by (2.11)} \\ &= \frac{P(S_t = \nu_m, \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)} | \lambda)}{\sum_{w \in \delta} P(S_t = w, \mathbf{S}_n^{(-t)} = \mathbf{s}_n^{(-t)} | \lambda)} \quad \text{..... by (2.9).} \end{aligned}$$

Both the numerator and denominator of this expression can be computed using the evaluation calculation described in Section 3.1. The calculations for continuous signal observations follow similarly with probabilities replaced by densities.

The second type of residuals which can be used to test the adequacy of a fitted HMM are the forecast pseudo-residuals. These are calculated from the conditional distributions given all preceding observations. That is, for continuous observations the normal pseudo-residual is defined as

$$z_t = \Phi^{-1}[P(S_t \leq s_t | \mathbf{S}_{t-1} = \mathbf{s}_{t-1})] \,.$$

The intervals $[z_t^-; z_t^+]$ for HMMs with discrete observations follow similarly,

$$\begin{aligned} z_t^- &= \Phi^{-1}[P(S_t < s_t | \mathbf{S}_{t-1} = \mathbf{s}_{t-1})] \\ z_t^+ &= \Phi^{-1}[P(S_t \leq s_t | \mathbf{S}_{t-1} = \mathbf{s}_{t-1})] \,. \end{aligned}$$

Equation (3.20) can be called upon to calculate these conditional probabilities.

Finally, a worked example of pseudo-residual analysis is provided in Section 6.3.1 of [46]. This analysis is done for various HMMs which have been fitted to the earthquake data which was discussed in Section 2.2 of this dissertation.

5.1.3 Performing Out-of-Time and Out-of-Sample Tests

In addition to using pseudo-residual analysis, out-of-time and out-of-sample tests may also aid in assessing the adequacy of a HMM which has been fit and in selecting a final HMM from various possible HMMs. The forecasting accuracy of a fitted HMM can also be assessed through these tests.

The out-of-time test is defined as follows. Suppose that observations for n time points have been observed. Let $x < n$ be the number of time points on which the test should be performed. Then, using the observed signals from the first $n - x$ time points, the parameters of the HMM can be estimated. Using the fitted HMM, the signals for the final x time points can be forecast (through the use of equation (3.20)) and compared to the actual signals which were observed.

While out-of-time tests may prove useful in some applications, it is important to note that observed signals are dependent on both the stochastic process of the hidden states and the probability distribution for each state through which the signals are emitted. Hence there are two sources of possible variability, and this may in turn lead to additional variability (than what would usually be expected for an out-of-time test) between the observed and forecasted signals.

When multiple observation sequences have been observed, out-of-sample tests can also be performed. To perform this, assume that y independent observation sequences have been observed and that $v < y$ of these sequences are randomly selected and used to estimate the model parameters. The remaining $y - v$ observation sequences can

then be used to assess the model (e.g. perform the model selection and adequacy tests mentioned in Section 5.1.1 and 5.1.2 using the remaining $y-v$ observation sequences). In this way the data used to assess the model is distinct from the data which was used to estimate the model parameters.

5.2 Adaptations of the Hidden Markov Model

A notable advantage of the HMM is its flexibility in that it can be modified or generalized depending on the application. This section will highlight some of the HMM adaptations discussed in the literature (see for example [46]). One such extension of the HMM, that of allowing direct dependencies among the signals, is discussed in greater detail in the next chapter of this dissertation.

Some degree of flexibility has already been detailed in this dissertation. For example, discussions have already been provided regarding how the HMM can easily be adapted to cater for signals being emitted according to explicitly defined probabilities for each state (the general HMM), or signals being emitted according to familiar probability distributions (the distribution HMMs). Examples are given in [46] as to when certain distributions may be applicable. To this end the following is noted:

- The Bernoulli distribution is appropriate for HMMs with binary counts (signals). Examples of such HMMs, given in [46], include daily rainfall occurrence (rain or no rain), consecutive departures of aeroplanes at an airport (on time, not on time) and daily trading of shares (traded or not traded).

- The Poisson and negative binomial distributions may be used for HMMs with unbounded discrete counts. In particular Section 2.2 discussed how a Poisson-HMM can improve the fit, when compared to an independent Poisson mixture model, for overdispersed data. It is further noted in [46] that a negative binomial HMM "may sensibly be used if even a Poisson-HMM seems unable

to accommodate the observed overdispersion". Examples of application given in [46] for Poisson- and negative binomial-HMMs include series of counts for breakdowns of technical equipment, earthquakes, insurance claims, accidents reported, products sold and defective items produced.

- Binomial-HMMs may be used to model series of bounded discrete counts. To this end define n_t as the number of trials at time t and x_t as the number of successes at time t. An example of a series of bounded counts, as given in [46], is purchasing preference (n_t = number of purchases of all brands on day t; x_t = number of purchases of a specific brand on day t). Binomial-HMMs are used in [4] and [27] to model the number of defaults for a credit portfolio (n_t = number of performing companies or accounts making up the portfolio at time t; x_t = number of performing companies or accounts at time t which default within a given time period, eg. a year). Additionally in [4] it is assumed that transitions between the hidden states of the HMM are not only driven by the Markov property, but also by macro-economic drivers. In both [4] and [27] it is assumed that while n_t is time dependent (i.e the portfolio size will vary over time as new companies or accounts enter the portfolio and defaults exit the portfolio) n_t is known for the time points for which defaults have been observed. An additional consideration if one requires that the forecast distribution of x_{T+h} be calculated (where $T + h$ is h time points after the last defaults have been observed) is that n_{T+h} must then either be assumed or estimated.

- A distribution HMM may also be used when signals from a continuous distribution are observed. In these instances the state-dependent signal distributions, which are assumed to generate the signals at each time point, are assumed to be continuous probability density functions. In particular exponential, Gamma and normal distributions are mentioned in [46], as well as an application of the normal-HMM in modelling share return series.

In addition to the above, [46] also illustrates how the HMM can be generalised to incorporate more complex types of observations; for example, at each time point a series of signals is emitted according to some multivariate distribution such as the multinomial distribution.

Another possible adaptation of the HMM is to allow covariates to be introduced into the model, either via the state transition probabilities (see for example [4] and [46]) or the signal probabilities / signal distributions (see for example [35]). These covariates allow models to incorporate time trend and seasonality components and also allow for the inclusion of other factors which may be of interest, e.g. economic conditions. By taking appropriate transformations into account, this can be achieved through the use of regression. In this way the state transition probabilities / signal probabilities (general HMM) / parameters of the state-dependent signal distributions (distribution HMM) will change through time as the covariates evolve through time. An application of this is given in [4] where movements in credit market conditions are modelled by HMMs. In this case the state transition probabilities are regressed to economic macro factors by making use of the logistic transformation.

The final adaptation of the HMM which will be discussed is that of incorporating additional dependencies into the model. To begin consider the underlying state process where a first order Markov chain has thus far been assumed. A generalisation of this for the HMM and the double chain Markov model (this model is detailed in the next chapter) described the literature is to replace the underlying first-order Markov chain by a higher order Markov chain (see for example [11], [22] and [46]). In particular, the time homogeneous state transition probabilities for a second-order Markov chain are as follows:

$$\begin{aligned} p_{i,j,k} &= P[X_{m+l} = k | X_1 = i_1, \ldots, X_{m-1} = i, X_m = j] \\ &= P[X_{m+l} = k | X_{m-1} = i, X_m = j] \\ &\text{for states } i_1, \ldots, i_{m-2}, i, j, k \in S, \text{ and } l \in \{1, 2, \ldots\}. \end{aligned}$$

A possible downside to using higher order models to describe the state process is that the number of parameters in the model can increase quite rapidly (i.e an increased number of transition probabilities need to be considered). To overcome this several papers (see for example [11], [22] and [39]) incorporate mixture transition distribution (MTD) models into the Markov chain, HMM or double chain Markov model framework in order to estimate the higher order transition probabilities. In short, MTD models provide a framework to approximate higher order transition probabilities through a defined model rather than estimate each individual transition probability directly. In this way the number of parameters which need to be estimated are reduced.

Up until this point, the only considered dependence between observations has been that which arises from the underlying state process. Additional dependencies in the observed signal process may however also be considered. In [46] an extension of the general HMM is discussed whereby the observed signal depends not only on the current state but also on the state at the previous time point. There may also be applications where direct dependence between the emitted signals is suspected, and should therefore be incorporated into the model. Thus the probability of observing a signal at some time point is dependent on both the state occupied at that time point and the previous emitted signal(s). One way of incorporating this into the HMM is to assume an autoregressive process for the observations (see [37], [46]).

Another possibility discussed in the literature is to assume that the signal process also possesses the Markov property. That is, both the state and signal processes are driven by the Markov property, where the signal process is also dependent on the states visited by the state process. This model is commonly referred to by the literature as the double-chain Markov model and it is this model which will be detailed in the next chapter.

Chapter 6

The Double-Chain Markov Model

6.1 Defining the Double-Chain Markov Model

Adaptations of the HMM were discussed in Section 5.2 of this dissertation. One such adaptation, namely the Double-Chain Markov model (DCMM) will be detailed in this chapter. The DCMM will first be introduced before model details, estimation and prediction will be discussed. The work presented in this chapter is adapted predominately from [10], [11], [22] and the previous chapters of this dissertation (since the DCMM is an extension of the HMM it will be shown that certain mathematics from the HMM can be extended to the DCMM).

6.1.1 Introducing the Double-Chain Markov Model

Markov chains and HMMs have been reviewed in the previous chapters of this dissertation. Recall that the Markov chain is a stochastic process where transitions between successive outputs of a discrete time random variable is governed by the Markov property. This process is entirely observable as each observed output is exactly identified with one state of the process. This was depicted in Figure 1.1 of Section 1.1. While Markov chains are widely used, there are applications where the model is not appro-

priate. For example in speech recognition there is not perfect identification between the state at a given point and the signal output. Instead, at each time point the state of the chain is unknown, but the output of another variable (the distribution of which depends entirely on the state of the model at the time point in question) is observed. This process is of course the HMM which has been discussed in detail. Importantly, the outputted signal sequence of the HMM is governed by the state process (whereby the state process is in turn governed, similar to the Markov chain, by the Markov property). It should however be noted that if the state process is not known / not assumed, then the probability of observing a given signal at some arbitrary time point k is dependent on the previous outputted signals.[1] However, given the state at time k, the probability of observing a given signal at time k is conditionally independent of all previous outputted signals. This was summarised in Figure 1.2 and is also made clear by the two equations below:

$$\begin{aligned} P(S_k = s_k | \mathbf{S}_{k-1} = \mathbf{s}_{k-1}) &\neq P(S_k = s_k) \\ P(S_k = s_k | X_k = i, \mathbf{S}_{k-1} = \mathbf{s}_{k-1}) &= P(S_k = s_k | X_k = i)\,. \end{aligned} \tag{6.1}$$

This conditional independence between the outputs of the HMM (equation (6.1)) may not always be justified. In fact in the literature there are numerous examples of processes governed by the HMM structure, but where the assumption of conditional independence between outputs is deemed to not be appropriate (see for example [10], [11], [22], and [46]). In particular, if it is assumed that successive outputs are related through the Markov property, then the resulting model is the double-chain Markov model (DCMM) presented in this chapter. That is, the DCMM has a similar stochastic framework to the HMM, but now it is assumed that for a given time point the signal emitted is not only dependant on the current hidden state, but also dependant (through the Markov property) on the previous observed signal (see Figure 1.3).

[1] The reason for this is that the observed signal sequence holds valuable information in predicting the state sequence, which in turn drives the signal process. Hence if the state process is not known / not assumed, then the previous outputted signals hold valuable information in calculating the probability of observing a given signal.

The name double-chain Markov model is now clear as the model is a combination of two inter-linked Markov chains; the hidden chain governing the relation between the states and the observable chain governing (together with the hidden state process) the relation between the observed outputs or signals.

The dependence of a signal on both the current state and the previous emitted signal can be explained as follows. The signal process can be considered as a Markov chain, but where the transition probability matrix is dependant on the current state occupied. That is, a signal transition probability matrix is associated with each state in the state space, and each time the DCMM enters a new state, the signal transition probability matrix for that state is used to determine which signal (given the previous signal) will be emitted for that time point. The output of the DCMM can thus be viewed as a time inhomogeneous Markov chain, where the transition probability matrix used for the outputs is driven by the state process of the DCMM.

A benefit of the DCMM is that the advantages of both the Markov chain and HMM are conserved - that is the system is driven by an unobserved latent process while successive outputs are dependent through the Markov property.

As an example of the DCMM, consider the following application adapted from [10]. In this application it is desired to model a time-series of daily average wind speeds at a specific location over a 17 year period. These wind speeds are of interest in order to determine the possible use of wind power in the area. More specifically two extreme conditions which can prevent a good exploitation of power need to be considered: days with exceptionally low wind speed and days with exceptionally high wind speed. Accordingly the data is classified into three categories, 'low wind speed', 'normal wind speed' and 'high wind speed'. Let these categories be denoted by C_l, C_n and C_h respectively.

Several models were used to model this data including Markov chains, HMMs and DCMMs. For each of these models, let $\{C_l, C_n, C_h\}$ represent the set of possible ob-

servations.

For the Markov chain, the output of the process is its state (that is the state process is also the signal process) and hence $\{C_l, C_n, C_h\}$ represents the state/signal space of the Markov chain. A single transition matrix is then used throughout to model transitions between these outputs $\{C_l, C_n, C_h\}$. That is, dependence between the wind speeds on successive days is modelled, but no underlying latent factor is considered.

The HMM considers an underlying latent factor by including the hidden state process. This could for example be some seasonal factor; at certain times of the year higher wind speeds could be expected, while at other times lower wind speeds could be the norm. Since the output of the HMM is a signal determined by its hidden state process, $\{C_l, C_n, C_h\}$ now represents the signal space of the HMM. And so the current state occupied would then influence the probability of observing one of the signals from $\{C_l, C_n, C_h\}$. While the HMM does incorporate this latent factor which drives the wind speed which is observed, it is assumed that there is no direct dependence between the wind speeds on successive days.

The DCMM incorporates these two models and conserves the advantage of each model. The DCMM once again incorporates the hidden state process (e.g. the process of the seasonal factor) which influences the signal which is observed from the signal space $\{C_l, C_n, C_h\}$.[2] However now direct dependence (through the Markov property) between the wind speeds on successive days is also modelled. This is done by estimating a separate signal transition probability matrix for each hidden state (seasonal factor) in the state space. In the application given in [10] it was found, using BIC as a model selection criterion, that of the models considered, the DCMM with two states was the most appropriate. The state transition matrix was estimated to be

$$\mathbf{P} = \begin{pmatrix} 0.9875 & 0.0125 \\ 0.0148 & 0.9852 \end{pmatrix}.$$

[2]For the DCMM, the output is a signal (influenced by its hidden state process), and so $\{C_l, C_n, C_h\}$ represents the signal space of the DCMM.

Thus, as intuitively expected, the seasonal factor is estimated to be quite stable through time (as the underlying data represents daily intervals).

The signal transition probability matrix for each state was estimated as

$$\mathbf{B}^{(1)} = \begin{pmatrix} 0.3550 & 0.6450 & 0 \\ 0.0805 & 0.8874 & 0.0321 \\ 0.0228 & 0.7721 & 0.2051 \end{pmatrix} \qquad \mathbf{B}^{(2)} = \begin{pmatrix} 0.1973 & 0.7846 & 0.0181 \\ 0.0361 & 0.8137 & 0.1502 \\ 0 & 0.6826 & 0.3174 \end{pmatrix}$$

where $\mathbf{B}^{(1)}$ represents the signal transition probability matrix when the DCMM is in state 1, and

$\mathbf{B}^{(2)}$ represents the signal transition probability matrix when the DCMM is in state 2.

It can be seen that transitions into C_l (column 1) are more likely using $\mathbf{B}^{(1)}$ than $\mathbf{B}^{(2)}$. Conversely, transitions into C_h (column 3) are more likely using $\mathbf{B}^{(2)}$ than in $\mathbf{B}^{(1)}$. This suggests that state 1 corresponds to seasons or time periods when lower wind speeds would be expected, while state 2 corresponds to seasons or time periods when higher wind speeds would be expected. For the DCMM, this then highlights the dependence of the output signal on both the previous signal (through the Markov property) and the current state (which is governed by the hidden Markov chain). The DCMM may thus prove particularly useful when it is expected that the transition probability matrix of a Markov chain could potentially change through time according to changes through time of some underlying latent process.

Based on the above discussion it may well be expected that both the time-homogeneous Markov chain and HMM are special cases of the DCMM. This desirable property does indeed hold true and is formally proven in Appendix A.

While the advantages of the DCMM have been mentioned, one notable disadvantage is that the DCMM will contain more parameters than either the Markov chain or the HMM. For a given application, these parameters will typically have to be estimated from the data observed (parameter estimation for the DCMM will be discussed later

in this chapter). Thus for a given application, one of the considerations which needs to be taken into account when assessing the suitability of the DCMM over the time-homogeneous Markov chain and the HMM is the amount of data which is available. The number of parameters which need to be estimated for each model is given below (where M represents the number of states in the state space and K represents the number of signals in the signal space):

- For the Markov chain, the number of parameters which need to be estimated is $(M-1)+M(M-1)$.
- For the HMM, the number of parameters which need to be estimated is $(M-1)+M(M-1)+M(K-1)$.
- For the DCMM, the number of parameters which need to be estimated is $(M-1)+M(M-1)+MK(K-1)$.

Finally, it should be noted that this dissertation will focus on the discrete-time, discrete-state and discrete-signal DCMM where the state transition probability matrix and the signal transition probability matrix for each state are assumed time homogeneous.

6.1.2 Model Assumptions and Notation

Now that the framework of the DCMM and its relation to the HMM has been outlined, model assumptions and notational changes for the DCMM will be formalised in this section.

To begin, recall that due its model structure the following assumptions (for $k \geq 1$) could be made for the HMM (see equations (2.1) - (2.6) from Section 2.1.3):

$$
\begin{aligned}
P[X_{k+t} = j \,|\, S_1, X_1, \ldots, S_k, X_k = i] &= P[X_{k+t} = j \,|\, X_k = i] \\
P[S_k = \nu_m \,|\, S_1, X_1, \ldots, S_{k-1}, X_{k-1}, X_k = i] &= P[S_k = \nu_m \,|\, X_k = i] \;=\; b_{im} \\
P[S_{k+t} = \nu_m \,|\, S_1, X_1, \ldots, S_k, X_k = i] &= P[S_{k+t} = \nu_m \,|\, X_k = i] \\
P[S_{k+t}, \ldots, S_n \,|\, S_1, \ldots, S_k, X_1, \ldots X_k = i] &= P[S_{k+t}, \ldots, S_n \,|\, X_k = i] \\
P[S_{k+t}, \ldots, S_n \,|\, S_1, \ldots, S_k, X_1, \ldots, X_n] &= P[S_{k+t}, \ldots, S_n \,|\, X_{k+t}, \ldots, X_n] \,.
\end{aligned}
\tag{6.2}
$$

The model assumptions for the DCMM follow from the above HMM assumptions, by taking the following into account. Firstly, recall that it is assumed that the HMM process begins at time 1. For the DCMM, since S_1 will depend on the output at the previous time point, an initial output at time 0 will be considered for the DCMM with no corresponding hidden state (as is shown in Figure 1.3). Secondly, since the state process of the DCMM follows the Markov property, and is in no way influenced by the observed signals, the first equation of (6.2) will still hold for the DCMM. Finally, due to the signal process being governed by both the Markov property and the current state, the remaining conditional probabilities of (6.2) will show additional dependence. Thus, the probabilities of equation (6.2) are expressed for the DCMM, where $k \geq 1$, as follows:

$$
\begin{aligned}
P[X_{k+t} = j \,|\, S_0, S_1, X_1, \ldots, S_k, X_k = i] &= P[X_{k+t} = j \,|\, X_k = i] \\
P[S_k = \nu_m \,|\, S_0, S_1, X_1, \ldots, S_{k-1} = \nu_j, X_{k-1}, X_k = i] &= P[S_k = \nu_m \,|\, S_{k-1} = \nu_j, X_k = i] \\
P[S_{k+t} = \nu_m \,|\, S_0, S_1, X_1, \ldots, S_k = \nu_j, X_k = i] &= P[S_{k+t} = \nu_m \,|\, S_k = \nu_j, X_k = i] \\
P[S_{k+t}, \ldots, S_n \,|\, S_0, S_1, \ldots, S_k = \nu_j, X_1, \ldots X_k = i] &= P[S_{k+t}, \ldots, S_n \,|\, S_k = \nu_j, X_k = i] \\
P[S_{k+t}, \ldots, S_n \,|\, S_0, S_1, \ldots, S_k, X_1, \ldots, X_n] &= P[S_{k+t}, \ldots, S_n \,|S_k,\, X_{k+t}, \ldots, X_n] \,.
\end{aligned}
\tag{6.3}
$$

For notational ease, under the previously mentioned time-homogeneous assumption, define the conditional probability of observing a signal for the DCMM as

$$b_{jm}^{(i)} = P[S_k = \nu_m \,|\, S_{k-1} = \nu_j, X_k = i]\,,$$

where $i \in S$; $\nu_j, \nu_m \in \delta$; $k \in \{1, 2, \ldots, n\}$.

Given the structure of the DCMM, it should be clear that initial state probabilities and the one-step state transition probabilities for the DCMM are subject to the same constraints which were applicable to the Markov chain and HMM (see equations (1.2) and (1.3)); while the signal probabilities for the DCMM are subject to:

$$\sum_{v_l \in \delta} b_{jl}^{(i)} = 1, \qquad \text{for } i \in S \text{ and } \nu_j \in \delta$$

$$b_{jm}^{(i)} \geq 0, \qquad \text{for } i \in S \text{ and } \nu_j, \nu_m \in \delta.$$

For a given state $i \in S$, let the matrix $\mathbf{B}^{(i)}$ contain the conditional signal transition probabilities for state i, where the (j, m) entry of $\mathbf{B}^{(i)}$ is $b_{jm}^{(i)}$. That is, if the DCMM is currently in state i, $\mathbf{B}^{(i)}$ will be the signal transition probability matrix used to determine the signal output at the current time point given the signal which was outputted at the previous time point. From this it can be seen that the output of the DCMM can be viewed as a time inhomogeneous Markov chain, where the transition probability matrix used for the outputs is dependent on the state of the DCMM (as previously mentioned).

6.1.3 Deriving Important Equations for the Double-Chain Markov Model

The mathematical particulars of the HMM have been discussed in detail in previous chapters. In particular, three important equations formed the foundation for the HMM, namely the forward, backward and Viterbi equations. These equations can be similarly defined for the DCMM, a difference however being that the sequence of

observed signals, $\mathbf{S}_k$ (where $k \leq n$), now includes S_0, the signal observed at time 0. These equations will once again form the foundation of the DCMM. As such computational forms of these equations are desired and will be discussed in section.

To begin, the forward equation for the DCMM is defined, similar to the HMM, as follows

$$F_k(j) = P(\mathbf{S}_k = \mathbf{s}_k, X_k = j|\lambda)\,,$$

where $j \in S$ and $k \in \{1, \ldots, n\}$.

Using a similar approach to that followed in Section 2.3.1, but replacing the assumptions from equation (6.2) with those in equation (6.3), it can easily be verified that

$$F_k(j) = b^{(j)}_{s_{k-1},s_k} \sum_{i \in S} F_{k-1}(i)\, p_{ij}\,,$$

where

$$\begin{aligned} F_1(j) &= P(S_0 = s_0, S_1 = s_1, X_1 = j|\lambda) \\ &= P(S_1 = s_1|S_0 = s_0, X_1 = j, \lambda)P(S_0 = s_0, X_1 = j|\lambda) \quad \text{.....} \quad \text{by (2.11)} \\ &= b^{(j)}_{s_0,s_1}\, P(S_0 = s_0, X_1 = j|\lambda)\,. \end{aligned}$$

Since S_0 and X_1 are independent and S_0 has been observed and is therefore known, it follows that

$$\begin{aligned} F_1(j) &= b^{(j)}_{s_0,s_1}\, P(S_0 = s_0|\lambda)P(X_1 = j|\lambda) \\ &= b^{(j)}_{s_0,s_1}\, .1.\, P(X_1 = j|\lambda) \\ &= b^{(j)}_{s_0,s_1}\, p_j\,. \end{aligned}$$

Next the backward equation for the DCMM is defined as follows:

$$B_k(i) = P(S_{k+1} = s_{k+1}, \ldots, S_n = s_n|S_k = s_k, X_k = i, \lambda)\,,$$

where $i \in S$ and $k \in \{1, \ldots, n-1\}$.

Note the extra term $S_k = s_k$ which did not appear in the backward equation of the HMM (equation (2.17)). However, equation (2.21) confirms that the backward equation for the HMM could have equivalently been expressed in this form.

Using a similar approach to that used in Section 2.3.2, but replacing the assumptions from equation (6.2) with those in equation (6.3), it can easily be verified that

$$B_k(i) = \sum_{j \in S} b^{(j)}_{s_k, s_{k+1}} \, B_{k+1}(j) \, p_{ij} \, .$$

As with the HMM, the backward equation for the DCMM at time n is set to 1 for each state in the state space; that is $B_n(i) = 1$ for each $i \in S$. This ensures that $0 \leq B_k(i) \leq 1$ for each $i \in S$ and each $k = 1, 2, \ldots, n$, which of course needs to hold true since $B_k(i)$ is a probability.

Finally, the Viterbi equation for the DCMM is defined, similar to the HMM, as follows

$$V_k(j) \;\; = \;\; \max_{i_1, \ldots, i_{k-1}} P\{\mathbf{X}_{k-1} = (i_1, \ldots, i_{k-1}), X_k = j, \mathbf{S}_k = \mathbf{s}_k | \lambda\} \, ,$$

where $k \in \{1, \ldots, n\}$ and $j, i_h \in S$ for $h = 1, \ldots, k-1$.

Using a similar approach to that used in Section 2.3.3, but replacing the assumptions from equation (6.2) with those in equation (6.3), it can easily be verified that

$$V_k(j) = b^{(j)}_{s_{k-1}, s_k} \, \max_{i \in S} \, \{p_{ij} \, V_{k-1}(i)\} \, ,$$

where

$$V_1(j) = P(X_1 = j, S_0 = s_0, S_1 = s_1 | \lambda) = F_1(j) = b^{(j)}_{s_0, s_1} \, p_j \, .$$

The above discussed iterative relationships greatly simplifies the computations of the forward, backward and Viterbi equations when calculations of their values are required, as will be the case in the next section.

6.2 Solving Problems Regarding the Double-Chain Markov Model

Estimation particulars regarding the DCMM will typically be of interest during applications of the DCMM. These include:

- The estimation of the likelihood of a sequence of signals $s_o, s_1, \ldots, s_n$ given a DCMM λ. This is commonly referred to in the literature as the evaluation problem.

- The estimation of the optimal sequence of hidden states given a DCMM λ and the sequence of observed signals. This is commonly referred to in the literature as the decoding problem.

- The estimation of the parameter set of a DCMM (the initial state probabilities, the state transition probability matrix, and the signal transition probability matrix for each state) given the sequence of observed signals. This is commonly referred to in the literature as the learning problem.

- Estimating the probabilities of future states which will be visited and future signals which will be emitted by the DCMM, given the sequence of observed signals.

This section will detail the estimation particulars of the above for the DCMM.

To begin, consider the evaluation problem where the problem of interest is to calculate the probability of the signal sequence $s_o, s_1, \ldots, s_n$ given a DCMM λ:

$$P(S_0 = s_o, S_1 = s_1, \ldots, S_n = s_n|\lambda) = P(\mathbf{S}_n = \mathbf{s}_n|\lambda)\,.$$

This can be calculated in one of three ways for the DCMM:

$$
\begin{aligned}
P(\mathbf{S}_n = \mathbf{s}_n|\lambda) &= \sum_{i\in S} F_n(i)\,, \quad \text{or} \\
P(\mathbf{S}_n = \mathbf{s}_n|\lambda) &= \sum_{i\in S} b^{(i)}_{s_o,s_1}\, B_1(i)\, p_i\,, \quad \text{or} \\
P(\mathbf{S}_n = \mathbf{s}_n|\lambda) &= \sum_{i\in S} F_k(i)B_k(i)\,, \qquad (6.4)
\end{aligned}
$$

where the forward and backward equations for the DCMM can be calculated using the techniques discussed in Section 6.1.3. The above three equations for the evaluation probability are derived using a similar approach to that which was used for the HMM (see Section 3.1.2).

In order to solve the decoding problem (i.e. optimally determine the sequence of hidden states which have been visited, given a DCMM λ and the sequence of observed signals), let $\hat{X}_k$ denote the optimal estimator of X_k, the hidden state of the DCMM at time k.

Recall from Section 3.2.2 that two approaches to solving the decoding problem for the HMM were considered; that of calculating $\hat{X}_k$ independently for each time point and that of treating the entire state sequence as a single entity which must be optimised. These two approaches can once again be followed for the DCMM.

Similar to the HMM note that

$$
\begin{aligned}
P(X_k = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) &= \frac{P(\mathbf{S}_n = \mathbf{s}_n, X_k = i|\lambda)}{P(\mathbf{S}_n = \mathbf{s}_n|\lambda)} \\
&= \frac{F_k(i)B_k(i)}{\sum\limits_{j\in S} F_k(j)B_k(j)}\,.
\end{aligned}
$$

Since $\sum\limits_{j\in S} F_k(j)B_k(j)$ is constant for each $i \in S$, it follows that,

$$
\begin{aligned}
\hat{X}_k &= \arg\max_{i\in S}\{P(X_k = i|\mathbf{S}_n = \mathbf{s}_n, \lambda)\} \\
&= \arg\max_{i\in S}\{F_k(i)B_k(i)\}
\end{aligned}
$$

for each $k = 1, 2, \ldots, n$.

As was the case with the HMM, while this approach will maximize the number of individually correct states, the 'optimal' state sequence estimated may not always be attainable. To see this, suppose that $p_{ij} = 0$ for some $i, j \in S$. This approach cannot guarantee that $\hat{X}_k = i$ and $\hat{X}_{k+1} = j$ (for some $k = 1, 2, \ldots, (n-1)$) will not occur in the 'optimal' state sequence. Similarly, $p_{ij} = 1$ may also result in an unattainable 'optimal' state sequence if this approach is used.

For the HMM, the solution to this which was presented in Section 3.2.2 was to regard the entire state sequence as a single entity. That is, the solution to the decoding problem will be the state sequence $(\hat{X}_1, \ldots, \hat{X}_n)$ such that

$$(\hat{X}_1, \ldots, \hat{X}_n) = \arg \max_{(i_1, \ldots, i_n)} \{P(\mathbf{X}_n = (i_1, \ldots, i_n) | \mathbf{S}_n = \mathbf{s}_n, \lambda)\} \ ,$$

where the likelihood which is maximised contains the entire state sequence.

To calculate the sequence $(\hat{X}_1, \ldots, \hat{X}_n)$ for the DCMM, the Viterbi algorithm can once again be performed in the same way which was described for the HMM, this time using the Viterbi equation for the DCMM (see Section 6.1.3). This can be proven using similar techniques to those used in the proof of the theorem 1 in Section 3.2.2, as is briefly discussed below.

To begin, notice that by equation (2.11)

$$P(\mathbf{X}_n = (i_1, \ldots, i_n) | \mathbf{S}_n = \mathbf{s}_n, \lambda) = \frac{P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda)}{P(\mathbf{S}_n = \mathbf{s}_n | \lambda)} \, .$$

Since the calculation of $P(\mathbf{S}_n = \mathbf{s}_n | \lambda)$ does not depend on the state sequence which has been visited, the problem of interest is equivalent to finding the state sequence $(i_1, \ldots, i_n)$ which will maximise

$$P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n | \lambda) \, .$$

Using the assumptions of the DCMM (equation (6.3)) it can easily be verified (using similar techniques to those used in the derivation of equation (3.1)) that for a given

state sequence

$$P(\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n|\lambda) = b^{(i_1)}_{s_0,s_1}\, b^{(i_2)}_{s_1,s_2}\, \ldots\, b^{(i_n)}_{s_{n-1},s_n}\, p_{i_1}\, p_{i_1,i_2}\, p_{i_2,i_3}\, \ldots\, p_{i_{n-1},i_n}. \quad (6.5)$$

Now define $(\dot{X}_1, \dot{X}_2, \ldots, \dot{X}_n)$ be the estimated state sequence for the DCMM derived from the Viterbi equation. That is

$$\begin{aligned} \dot{X}_n &= \arg\max_{j\in S}\{V_n(j)\} \\ \dot{X}_k &= \psi_{k+1}(\dot{X}_{k+1}) \qquad\qquad \text{for } k = 1, \ldots, n-1\,, \end{aligned}$$

where

$$\begin{aligned} \psi_k(j) &= \arg\max_{i\in S}\{p_{ij}\, V_{k-1}(i)\} && \text{for each } j \in S \text{ and } k = 2, \ldots, n\,, \\ V_k(j) &= b^{(j)}_{s_{k-1},s_k} \max_{i\in S}\{p_{ij}\, V_{k-1}(i)\} && \text{for each } j \in S \text{ and } k = 2, \ldots, n\,, \\ V_1(j) &= b^{(j)}_{s_0,s_1}\, p_j && \text{for each } j \in S\,. \end{aligned}$$

It can then be verified, using similar techniques to those used in the proof of the theorem 1 in Section 3.2.2, that for the DCMM

$$\begin{aligned} & \max_{i_1,\ldots,i_n} P\{\mathbf{X}_n = (i_1, \ldots, i_n), \mathbf{S}_n = \mathbf{s}_n|\lambda\} \\ = \quad & b^{(\dot{X}_n)}_{s_{n-1},s_n}\, \ldots\, b^{(\dot{X}_2)}_{s_1,s_2}\, b^{(\dot{X}_1)}_{s_0,s_1}\, p_{\dot{X}_{n-1},\dot{X}_n}\, \cdots\, p_{\dot{X}_1,\dot{X}_2}\, p_{\dot{X}_1}\,. \end{aligned}$$

This then demonstrates the validity of Viterbi Algorithm in solving the decoding problem for the DCMM as it has been shown that

$$(\dot{X}_1, \dot{X}_2, \ldots, \dot{X}_n) = \arg\max_{(i_1,\ldots,i_n)}\{P(\mathbf{X}_n = (i_1, \ldots, i_n)|\mathbf{S}_n = \mathbf{s}_n, \lambda)\}\,.$$

Next the learning problem for the DCMM is discussed, that is estimating the parameter set of the DCMM given the sequence of observed signals. To this end, using an approach comparable to that used for HMMs, the Baum-Welch Algorithm (BWA) estimates for the DCMM parameters can be derived.

To begin, firstly define

$$\begin{aligned}
\xi_k(i,j) &= P(X_k = i, X_{k+1} = j | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
\gamma_k(i) &= P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
\gamma_{k,h}(i) &= \begin{cases} \gamma_k(i) & \text{if} \quad s_{k-1} = \nu_h \\ 0 & \text{if} \quad s_{k-1} \neq \nu_h \end{cases} \\
\gamma_{k,h,m}(i) &= \begin{cases} \gamma_k(i) & \text{if} \quad s_{k-1} = \nu_h \text{ and } s_k = \nu_m \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{6.6}$$

Using similar mathematics to that described for the HMM, but replacing the assumptions from equation (6.2) with those in equation (6.3), computational forms for $\xi_k(i,j)$ and $\gamma_k(i)$ can be derived for the DCMM:

$$\begin{aligned}
\xi_k(i,j) &= \frac{F_k(i)\, p_{ij}\, b^{(j)}_{s_k,s_{k+1}}\, B_{k+1}(j)}{P(\mathbf{S}_n = \mathbf{s}_n|\lambda)} \\
\gamma_k(i) &= \frac{F_k(i) B_k(i)}{P(\mathbf{S}_n = \mathbf{s}_n|\lambda)}\,,
\end{aligned}$$

where $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ can be calculated using equation (6.4).

The following interpretations can be made about the above probabilities:

$$\sum_{k=1}^{n-1} \gamma_k(i) \;=\; \text{expected number of transitions from state } i \text{ during the first } n \text{ observed time points,}$$

$$\sum_{k=1}^{n-1} \xi_k(i,j) \;=\; \text{expected number of transitions from state } i \text{ into state } j \text{ during the first } n \text{ observed time points,}$$

$$\sum_{k=1}^{n} \gamma_{k,h}(i) \;=\; \text{expected number of times, during the first } n \text{ time points, that the DCMM is in state } i \text{ when the previous emitted signal was } \nu_h\,,$$

$$\sum_{k=1}^{n} \gamma_{k,h,m}(i) \;=\; \text{expected number of times, during the first } n \text{ time points, that the DCMM is in state } i \text{ when the previous emitted signal was } \nu_h \text{ and the current signal emitted is } \nu_m\,.$$

Proofs for the first two equations were formally derived for the HMM in Appendix B and can be similarly derived for the DCMM. Proofs for the last two equations are shown in Appendix C.

Now define $\lambda^* = (\mathbf{P}^*, \mathbf{B}^*, \mathbf{a}^*)$ to be the current estimate of the parameters for the DCMM, and $\hat{\lambda} = (\hat{\mathbf{P}}, \hat{\mathbf{B}}, \hat{\mathbf{a}})$ to be the re-estimate of λ^*.

Also define

$$\gamma_k^*(i),\ \xi_k^*(i,j),\ \gamma_{k,h}^*(i) \text{ and } \gamma_{k,h,m}^*(i) \tag{6.7}$$

to be the values for $\gamma_k(i),\ \xi_k(i,j),\ \gamma_{k,h}(i)$ and $\gamma_{k,h,m}(i)$ calculated using λ^*.

Then for $i, j \in S$ and $\nu_j, \nu_m \in \delta$ the elements of $\hat{\lambda}$ can be calculated as follows:

$$\begin{aligned} \hat{p}_i &= P(X_1 = i | \mathbf{S}_n = \mathbf{s}_n, \lambda^*) \\ &= \gamma_1^*(i) \end{aligned} \tag{6.8}$$

$$\begin{aligned} \hat{p}_{ij} &= \text{proportion of times that, when the DCMM is in state } i, \text{ a transition} \\ &\quad \text{into state } j \text{ occurs} \\ &= \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \\ &= \frac{\sum_{k=1}^{n-1} \xi_k^*(i,j)}{\sum_{k=1}^{n-1} \gamma_k^*(i)} \end{aligned} \tag{6.9}$$

$$\begin{aligned} \hat{b}_{jm}^{(i)} &= \text{proportion of times that, when the DCMM is in state } i \text{ and the previous} \\ &\quad \text{signal emitted was } \nu_j, \text{ signal } \nu_m \text{ is emitted} \\ &= \frac{\{\text{expected number of times the DCMM is in state } i \text{ when the previous} \\ &\quad \text{emitted signal was } \nu_j \text{ and the current signal emitted is } \nu_m\}}{\{\text{expected number of times the DCMM is in state } i \text{ when the previous} \\ &\quad \text{emitted signal was } \nu_j\}} \\ &= \frac{\sum_{k=1}^{n} \gamma_{k,j,m}^*(i)}{\sum_{k=1}^{n} \gamma_{k,j}^*(i)} . \end{aligned} \tag{6.10}$$

The expressions (6.8)-(6.10), evaluated at the current parameter estimates, provide updated estimates of p_i, p_{ij} and $b_{jm}^{(i)}$.

At first glance it may appear that the Baum-Welch equations (6.8)-(6.9) for the DCMM will produce identical estimates to the Baum-Welch equations (4.7)-(4.8) for the HMM. This however will not be the case as the calculation of the forward and backward equations, which are used in the calculations of $\gamma_k^*(i)$ and $\xi_k^*(i,j)$, differ between the HMM and the DCMM.

When discussing the BWA for the HMM, several comments and findings were made. These can also be extended to the BWA for the DCMM as is discussed next.

Firstly it should be noted that the BWA for the DCMM is an example of the Expectation Maximization (EM) algorithm (the EM algorithm is detailed in Appendix B). That is the Baum-Welch re-estimation equations (equations (6.8)-(6.10)) are identical to the iteration steps which arise from the EM algorithm applied to this particular problem. The mathematics proving this for the DCMM are presented in Appendix C. This result is key as important properties have been proven for the EM algorithm (see Appendix B) and can thus be extended to the BWA estimates. For example, from the alignment of the BWA to the EM algorithm, the following can be concluded for the Baum-Welch estimates for the DCMM:

1) either λ^* defines a critical value of the likelihood function, $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$, in which case the above calculations will produce $\hat{\lambda} = \lambda^*$, or

2) model $\hat{\lambda}$ results in a higher value in the likelihood function than λ^* did - that is $P(\mathbf{S}_n = \mathbf{s}_n|\hat{\lambda}) > P(\mathbf{S}_n = \mathbf{s}_n|\lambda^*)$. Therefore a new model, $\hat{\lambda}$, has been found from which the observed signal sequence is more likely to have been produced.

Based on the above, if $\hat{\lambda}$ is iteratively used in place of λ^* in the re-estimation calculations, the probability of the observed signal sequence being produced by the estimated model is improved (until convergence is achieved). That is, the updated BWA estimates will result in the value of the likelihood function being repeatedly increased until some limiting point is reached. The final result of this re-estimation procedure is then the maximum likelihood estimator (this is formally proven for the EM algorithm in Appendix B). It should however be noted that the BWA only leads to a local maxima of the likelihood function, and that in most applications many local maxima are likely exist. This however is the most which can be achieved since, to the best of knowledge at the time of writing, no analytical or numerical methods exist in the literature which will solve for the global maxima of the likelihood $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$ for

the DCMM. Since only a local maxima can be found, the choice of the initial values of λ used to train the BWA will influence the final estimated values.

As was the case with the HMM, a pleasing property of the BWA is that at each iteration the parameter estimates satisfy the DCMM parameter constraints (provided of course that the initial estimates chosen for the BWA satisfy these constraints):

$$\sum_{i \in S} \hat{p}_i = 1$$

$$\hat{p}_i \geqslant 0, \qquad \text{for } i \in S, \text{ and}$$

$$\sum_{j \in S} \hat{p}_{ij} = 1, \qquad \text{for } i \in S$$

$$\hat{p}_{ij} \geqslant 0, \qquad \text{for } i, j \in S, \text{ and}$$

$$\sum_{v_m \in \delta} \hat{b}_{jm}^{(i)} = 1, \qquad \text{for } i \in S \text{ and } \nu_j \in \delta$$

$$\hat{b}_{jm}^{(i)} \geq 0, \qquad \text{for } i \in S \text{ and } \nu_j, \nu_m \in \delta.$$

And so the final estimated values of λ produced by the BWA will satisfy the DCMM parameter constraints.

The property that $\hat{p}_i \geqslant 0$, $\hat{p}_{ij} \geqslant 0$ and $\hat{b}_{jm}^{(i)} \geqslant 0$ for each iteration is guaranteed from the fact that forward and backward equations will be guaranteed to be greater than or equal to zero for each observed time point, provided that the initial estimates for p_i, p_{ij} and b_{jm} are chosen to be greater than or equal to zero (see Section 6.1.3 for details of this).

The remaining three properties can once again be proven by considering how the Baum-Welch re-estimation equations for the DCMM can be derived by making use of the EM algorithm. In Appendix C it is shown that these properties are guaranteed since, when deriving the re-estimation equations which will maximise $P(\mathbf{S}_n = \mathbf{s}_n|\lambda)$, Lagrange multipliers are used in the EM algorithm to ensure that these constraints

are satisfied for each iteration. Similar to the HMM, these three properties can also be proven algebraically by noting that if the partition rule for probability (equation (2.9)) is applied to equation (6.6), then $\sum_{j\in S} \xi_k(i,j) = \gamma_k(i)$ and $\sum_{\nu_m\in\delta} \gamma_{k,j,m}(i) = \gamma_{k,j}(i)$ is obtained, which when applied to equations (6.8)-(6.10) yield the desired properties.

In applications of the DCMM, multiple observation sequences may be available. To this end an adaptation of the BWA is required such that all available data is utilised in the estimation of the DCMM parameters. To begin, consider M independent observation sequences notated by

$$\acute{\mathbf{S}} = [\mathbf{S}_{n_1}^{(1)}, \mathbf{S}_{n_2}^{(2)}, ..., \mathbf{S}_{n_M}^{(M)}],$$

where $\mathbf{S}_{n_r}^{(r)} = (s_1^{(r)}, s_2^{(r)}, ..., s_{n_r}^{(r)})$ is the r^{th} observation sequence, consisting of n_r individual signals (observations), and $r \in \{1, 2, ..., M\}$.

It is then desired to use the entire set of data to train a single DCMM. Since all the observation sequences are independent, the likelihood is equal to

$$P(\acute{\mathbf{S}} \,|\, \lambda) \;=\; \prod_{r=1}^{M} P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \lambda)\,,$$

where $P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \lambda)$ is the likelihood of the r^{th} observation sequence and can be calculated using equation (6.4).

As was the case for a single observation sequence, define $\lambda^* = (\mathbf{P}^*, \mathbf{B}^*, \mathbf{a}^*)$ to be the current estimate of the parameter set for the DCMM, and $\hat{\lambda} = (\hat{\mathbf{P}}, \hat{\mathbf{B}}, \hat{\mathbf{a}})$ to be the Baum-Welch re-estimate of this parameter set.

Also, define

$$\gamma_k^{*(r)}(i),\ \xi_k^{*(r)}(i,j),\ \gamma_{k,h}^{*(r)}(i) \text{ and } \gamma_{k,h,m}^{*(r)}(i)$$

be the probabilities corresponding to those given in equation (6.6), calculated for the r^{th} observation sequence using λ^*, where $r = 1, 2, ..., M;\ k = 1, ..., n_r;\ i, j \in S$; and

$\nu_h, \nu_m \in \delta$.

Since the Baum-Welch re-estimation equations for a single observation sequence are based on the expected number of occurrences of certain events, it is suggested in [10] that the BWA re-estimation equations (equations (6.8) - (6.10)) be adapted as follows to take into account the information from the M sequences:

$$\begin{aligned}
\hat{p}_i &= \frac{1}{M}\sum_{r=1}^{M}\gamma_1^{*(r)}(i) \\
\hat{p}_{ij} &= \frac{\sum_{r=1}^{M}\sum_{k=1}^{n_r-1}\xi_k^{*(r)}(i,j)}{\sum_{r=1}^{M}\sum_{k=1}^{n_r-1}\gamma_k^{*(r)}(i)} \\
\hat{b}_{jm}^{(i)} &= \frac{\sum_{r=1}^{M}\sum_{k=1}^{n_r}\gamma_{k,j,m}^{*(r)}(i)}{\sum_{r=1}^{M}\sum_{k=1}^{n_r}\gamma_{k,j}^{*(r)}(i)} .
\end{aligned} \tag{6.11}$$

In Section 4.1.3 the approach used by [18] to extend the BWA for multiple observation sequences was discussed for the HMM. Similarly this approach is outlined for the DCMM next.

Under this approach it is suggested that the parameters of λ first be estimated using the single sequence Baum-Welch re-estimation equations (equations (6.8) to (6.10)) for each individual observation sequence. Thus M distinct estimates are obtained for each parameter. The final Baum-Welch estimates, for the multiple observation

sequences, are then given by:

$$\begin{aligned}
\hat{p}_i &= \sum_{r=1}^{M} \frac{W_r}{N_a}\, \hat{p}_i^{(r)} \\
\hat{p}_{ij} &= \sum_{r=1}^{M} \frac{W_r}{N_b}\, \hat{p}_{ij}^{(r)} \\
\hat{b}_{jm}^{(i)} &= \sum_{r=1}^{M} \frac{W_r}{N_c}\, \hat{b}_{jm}^{(i)(r)}\,, \qquad (6.12)
\end{aligned}$$

where $\hat{\lambda}^{(r)} = \left(\hat{\mathbf{P}}^{(r)}, \hat{\mathbf{B}}^{(r)}, \hat{\mathbf{a}}^{(r)}\right)$ is the final Baum-Welch estimate obtained from $\mathbf{S}_{n_r}^{(r)}$,
W_r is the weighting factor for the estimates from $\mathbf{S}_{n_r}^{(r)}$,
N_a, N_b and N_c are normalization factors.

Typically the weight factors used in the above calculations include unit weight factors ($W_r = 1$ for each observation sequence, that is the estimated parameters from each individual observation will have equal weight), weight factors expressed as a function of $P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \hat{\lambda}^{(r)})$ and weight factors expressed as a function of $P(\acute{\mathbf{S}} \,|\, \hat{\lambda}^{(r)})$. For each of these weightings, 'trimmed' weight factors can also be considered whereby the weight factors for unlikely models (as determined by either $P(\mathbf{S}_{n_r}^{(r)} = \mathbf{s}_{n_r}^{(r)} \,|\, \hat{\lambda}^{(r)})$ or $P(\acute{\mathbf{S}} \,|\, \hat{\lambda}^{(r)})$) are set to 0.

When applications consisting of multiple observation sequences are performed for the DCMM it is advised that both the approach described by equation (6.11) and the approach described by equation (6.12) be considered.

The final comment to be made regarding the BWA for the DCMM is that when calculating the BWA re-estimation equations, scaling may be required as the forward and backward equations can easily take on values too small to be handled by a computer. This is also noted in [10]. In Section 4.1.2.1 of this dissertation a technique for scaling the forward and backward equations for the HMM was discussed. This technique can also be used to scale the forward and backward equations for the DCMM.

Another aspect of the DCMM which may be of interest during applications is that of forecasting the state which will be visited and/or the signal which will be emitted at some future time point. A similar discussion was provided for the HMM in Section 3.4.3.

To begin assume that signals for the first n time points has been observed. Using similar mathematics to that which was used in Section 3.4.3, the forecasting distribution for the state visited at time $n+h$, where positive integer h is termed the forecast horizon, can be derived as the following:

$$P(X_{n+h} = j|\mathbf{S}_n = \mathbf{s}_n, \lambda) = \frac{1}{\sum\limits_{l \in S} F_n(l)} \sum_{i \in S} p_{ij}(h)\, F_n(i)\,, \tag{6.13}$$

where $\mathbf{P}^{(h)} = \{p_{ij}(h)\} = \mathbf{P}^h$ by equation (1.6).

While at first glance equation (6.13) may appear identical to the state forecasting distribution for the HMM (see equation (3.19)), this is not the case since the calculation of the forward equations differs for the HMM and DCMM.

Next the forecasting distribution for the signal emitted at time $n+h$ is derived. To begin consider

$$\begin{aligned}
& P(S_{n+1} = \nu_m|\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
= & \sum_{i \in S} P(S_{n+1} = \nu_m, X_{n+1} = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad ..... \quad \text{by (2.9)} \\
= & \sum_{i \in S} P(S_{n+1} = \nu_m|X_{n+1} = i, \mathbf{S}_n = \mathbf{s}_n, \lambda)\, P(X_{n+1} = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad ..... \quad \text{by (2.11)} \\
= & \sum_{i \in S} P(S_{n+1} = \nu_m|X_{n+1} = i, S_n = s_n, \lambda)\, P(X_{n+1} = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad ..... \quad \text{by (6.3)} \\
= & \sum_{i \in S} b^{(i)}_{s_n,m}\, P(X_{n+1} = i|\mathbf{S}_n = \mathbf{s}_n, \lambda)\,,
\end{aligned} \tag{6.14}$$

where $P(X_{n+1} = i|\mathbf{S}_n = \mathbf{s}_n, \lambda)$ can be calculated using equation (6.13);

$$
\begin{aligned}
& P(S_{n+2} = \nu_m|\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
= & \sum_{i\in S}\sum_{\nu_j\in\delta} P(S_{n+2} = \nu_m|X_{n+2} = i, S_{n+1} = \nu_j, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
& \quad \times P(X_{n+2} = i, S_{n+1} = \nu_j|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.9) and (2.11)} \\
= & \sum_{i\in S}\sum_{\nu_j\in\delta} P(S_{n+2} = \nu_m|X_{n+2} = i, S_{n+1} = \nu_j, \lambda) \\
& \quad \times P(X_{n+2} = i|S_{n+1} = \nu_j, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
& \quad \times P(S_{n+1} = \nu_j|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.11) and (6.3)} \\
= & \sum_{i\in S}\sum_{\nu_j\in\delta} b_{jm}^{(i)}\, P(X_{n+2} = i|\mathbf{S}_{n+1} = \mathbf{s}_{n+1}, \lambda)\, P(S_{n+1} = \nu_j|\mathbf{S}_n = \mathbf{s}_n, \lambda)\,,
\end{aligned}
$$

where it is assumed that $\mathbf{s}_{n+1} = (s_0, s_1, \ldots, s_n, \nu_j)$ and $P(X_{n+2} = i|\mathbf{S}_{n+1} = \mathbf{s}_{n+1}, \lambda)$ and $P(S_{n+1} = \nu_j|\mathbf{S}_n = \mathbf{s}_n, \lambda)$ can be calculated using equation (6.13) and equation (6.14) respectively;

$$
\begin{aligned}
& P(S_{n+3} = \nu_m|\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
= & \sum_{i\in S}\sum_{\nu_j\in\delta}\sum_{\nu_l\in\delta} P(S_{n+3} = \nu_m|X_{n+3} = i, S_{n+2} = \nu_j, S_{n+1} = \nu_l, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
& \quad \times P(X_{n+3} = i, S_{n+2} = \nu_j, S_{n+1} = \nu_l|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.9) and (2.11)} \\
= & \sum_{i\in S}\sum_{\nu_j\in\delta}\sum_{\nu_l\in\delta} P(S_{n+3} = \nu_m|X_{n+3} = i, S_{n+2} = \nu_j, \lambda) \\
& \quad \times P(X_{n+3} = i|S_{n+2} = \nu_j, S_{n+1} = \nu_l, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
& \quad \times P(S_{n+2} = \nu_j, S_{n+1} = \nu_l|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.11) and (6.3)} \\
= & \sum_{i\in S}\sum_{\nu_j\in\delta}\sum_{\nu_l\in\delta} b_{jm}^{(i)}\, P(X_{n+3} = i|\mathbf{S}_{n+2} = \mathbf{s}_{n+2}, \lambda) \\
& \quad \times P(S_{n+2} = \nu_j|S_{n+1} = \nu_l, \mathbf{S}_n = \mathbf{s}_n, \lambda)\, P(S_{n+1} = \nu_l|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \text{.....} \quad \text{by (2.11)} \\
= & \sum_{i\in S}\sum_{\nu_j\in\delta}\sum_{\nu_l\in\delta} b_{jm}^{(i)}\, P(X_{n+3} = i|\mathbf{S}_{n+2} = \mathbf{s}_{n+2}, \lambda) \\
& \quad \times P(S_{n+2} = \nu_j|\mathbf{S}_{n+1} = \mathbf{s}_{n+1}, \lambda)\, P(S_{n+1} = \nu_l|\mathbf{S}_n = \mathbf{s}_n, \lambda)\,,
\end{aligned}
$$

where it is assumed that $\mathbf{s}_{n+1} = (s_0, s_1, \ldots, s_n, \nu_l)$. and $\mathbf{s}_{n+2} = (s_0, s_1, \ldots, s_n, \nu_l, \nu_j)$.

The terms $P(X_{n+3} = i|\mathbf{S}_{n+2} = \mathbf{s}_{n+2}, \lambda)$, $P(S_{n+2} = \nu_j|\mathbf{S}_{n+1} = \mathbf{s}_{n+1}, \lambda)$ and $P(S_{n+1} = \nu_l|\mathbf{S}_n = \mathbf{s}_n, \lambda)$ can be calculated using equations (6.13) and equation (6.14).

And so the above procedure can be continued until the desired forecasting horizon is reached. It can be seen from the above that, due to the dependence structure within the output of the DCMM, quite detailed computations are required to calculate the probability of future signals being emitted for large h.

6.3 Additional Considerations for the Double Chain Markov Model

In addition to the discussions surrounding the DCMM presented in this chapter, further concepts which were discussed for the HMM in this dissertation can also be extended to the DCMM. These include

- deriving marginal distributions and moments for the DCMM,
- implementation considerations for the BWA for the DCMM,
- utilising direct maximisation of the likelihood function in order to estimate the model parameters (as opposed to using the Baum Welch / EM algorithm approach which was discussed for the DCMM in this chapter),
- Bayesian estimation for the DCMM,
- utilising standard errors and confidence intervals to assess the adequacy of estimated model parameters,
- assessing the fit of the estimated model to the observed data through model selection criterion and pseudo-residuals.

Comments surrounding these areas are similar to those made for the HMM and as such will not be repeated again here. One point to elaborate on however is a discussion presented in [23]. In [23] Bayesian estimation for the DCMM is explored. An algorithm for sampling from the posterior distribution associated with the DCMM, when multiple independent observation sequences are observed, is presented. Simulation studies and an application to real data (relating to credit rating migrations) are also presented in [23] to illustrate the proposed algorithm. This application will be further discussed in Chapter 8 of this dissertation.

Model adaptations to the DCMM which has been detailed in this chapter are also explored in the literature. In particular, a discussion on higher order DCMMs is presented in [11] and [22]. In these papers it is shown how the DCMM can be adapted to incorporate higher orders on both the process of the hidden states and the process of the observations. That is, for order g in the state process and order f in the signal process, the following assumptions are made:

$$\begin{aligned} & P[X_m = k | X_1 = i_1, S_1 = s_1, \ldots, X_{m-1} = i_{m-1}, S_{m-1} = s_{m-1}] \\ = \; & P[X_m = k | X_{m-1} = i_{m-1}, X_{m-2} = i_{m-2}, \ldots, X_{m-g} = i_{m-g}] \end{aligned}$$

and

$$\begin{aligned} & P[S_m = \nu_k | X_1 = i_1, S_1 = s_1, \ldots, X_{m-1} = i_{m-1}, S_{m-1} = s_{m-1}, X_m = i_m] \\ = \; & P[S_m = \nu_k | X_m = i_m, S_{m-1} = s_{m-1}, S_{m-2} = s_{m-2}, \ldots, S_{m-f} = s_{m-f}] \,. \end{aligned}$$

These assumptions will greatly increase the number of parameters in the model. For example if there are M states in the state space and K signals in the signal space, then the number of parameters which would need to be estimated would increase from

- $M - 1$ to $\sum_{l=0}^{g-1} M^l(M-1)$ for the initial state probabilities,
- $M(M-1)$ to $M^g(M-1)$ for the state transition probabilities,

- $MK(K-1)$ to $MK^f(K-1)$ for the signal transition probabilities.

In order to reduce the number of parameters for such a model, [11] and [22] propose the use of a Mixture Transition Distribution model (MTD), initially introduced by [39]. The MTD model was overviewed in Section 5.2 of this dissertation.

Applications comparing the performance of different dimension and different order Markov chains, HMMs and DCMMs (with and without the use of a MTD model) are also explored in [11], [22] and [23]. These studies will be explored in more detail in Chapter 8 of this dissertation.

Chapter 7

A Simulation Study of Hidden and Double Chain Markov Models

Previous chapters of this dissertation have provided theoretical discussions on HMMs and DCMMs. The sections of this chapter will explore how some of the aspects of these models actually perform in practice. After careful consideration it was decided that this analysis would best be facilitated through simulated data (as opposed to sourcing actual data) as this allows knowledge of the true model parameters and underlying state sequence. The approach used to perform these simulations are detailed in the relevant sections.

In order to preserve the structure and flow of each subsection, all graphs relating to a particular subsection are displayed at the end of the relevant subsection. Finally it should be noted that all code required to perform the simulations and analysis presented in this chapter was written entirely by the author of this dissertation.

7.1 Exploring the Baum-Welch Algorithm for the HMM

The BWA for the HMM has been detailed in previous discussions in this dissertation. The following key questions regarding the BWA for the HMM will now be explored in this section:

- How accurately does the BWA estimate the actual model parameters?
- What influence does the choice of the starting parameter values used to train the BWA have on the final parameters estimated by the BWA?
- What influence does the length of the signal sequence used to train the BWA have on the final parameters estimated by the BWA?
- How sensitive is the BWA to the signal sequence which is used to train it? That is, if different signal sequences, all simulated from a single HMM, are used to train the BWA, how variable will the different BWA estimates be?
- By considering the above points, can a technique be derived to enhance the effectiveness of the BWA when used in practice?

In exploring the above, it is believed that the techniques and findings presented in this section can greatly aid a practitioner in both understanding and implementing the BWA.

7.1.1 Exploring the Effect which Different Starting Parameter Values has on the Baum-Welch Algorithm

Simulations to test the effect that different starting parameter values have on the BWA for the HMM are now discussed. In particular, the structure of the simulations performed is first outlined before results are analysed.

To begin, the following HMM, $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$, was chosen to perform the simulations:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 0.70 & 0.30 \\ 0.15 & 0.85 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} 0.70 & 0.30 \\ 0.10 & 0.90 \end{pmatrix}.$$

Assuming that the parameters of λ are unknown, there are essentially five parameters which need to be estimated by the BWA namely a_1, p_{11}, p_{21}, b_{11}, b_{21} (since of course $a_2 = 1 - a_1$; $p_{12} = 1 - p_{11}$; $p_{22} = 1 - p_{21}$; $b_{12} = 1 - b_{11}$ and $b_{22} = 1 - b_{21}$; and as was discussed in Section 4.1.1 these properties are respected by the BWA). It was noted in Section 4.1.1 that the BWA estimate for a_1 will tend to either 0 or 1. For this reason the BWA estimate for a_1 will not be a focus of this simulation exercise. What is however of interest is how accurately the values for p_{11}, p_{21}, b_{11}, b_{21} are recovered by the BWA.

Using λ specified above, a HMM process was simulated. That is, an underlying state sequence and a corresponding signal sequence were simulated using the probabilities specified in λ. In order to ensure that meaningful results were achieved when performing the BWA, it was decided to ensure that the sequence lengths were suitably long (so as to ensure that there is sufficient data to train the BWA). To this end the simulated state and signal sequence each consisted of 5500 data points (shorter data sequences are also used later in this section to see how the BWA performs when less training data is available). Let the simulated state and simulated signal sequence be denoted by $\tilde{\mathbf{x}}_{5500}^{1}$ and $\tilde{\mathbf{s}}_{5500}^{1}$ respectively, where 5500 denotes the length of the state and signal sequences and 1 signifies that only one state and signal sequence has been simulated.

From the specified values of λ, it is noted that the limiting steady state probabilities for the underlying Markov chain is $\pi = [\frac{1}{3}, \frac{2}{3}]$. This can be confirmed through equation (1.9). Furthermore, the marginal probabilities of the HMM outputting signal 1 and signal 2 are 0.3 and 0.7 respectively (see equation (3.18) for details of this calcu-

lation). If simulated correctly, these probabilities should be reflected in the simulated state and signal sequence. This is indeed the case as state 1 and state 2 occur 1,807 and 3,693 times respectively in the simulated state sequence (a proportion of 0.33 and 0.67 respectively) and signal 1 and signal 2 occur 1,628 and 3,872 times respectively in the simulated signal sequence (a proportion of 0.30 and 0.70 respectively).

To quantify how accurately the BWA estimates the parameters of λ, 1,000 distinct starting values to train the BWA were generated, denoted by

$$\tilde{\lambda}^{1000} = \{\tilde{\lambda}^{(1)}, \tilde{\lambda}^{(2)} \ldots, \tilde{\lambda}^{(1000)}\}.$$

These were generated by drawing random numbers from the uniform(0,1) distribution, while still ensuring that the appropriate properties of λ are conserved (i.e. the constraints given in equations (1.2), (1.3) and (2.7)).

Next $\tilde{\mathbf{s}}^1_{5500}$ and $\tilde{\lambda}^{1000}$ were used to perform the BWA and create 1,000 distinct BWA estimates for λ, denoted as

$$\hat{\lambda}^{1000}_{5500} = \{\hat{\lambda}^{(1)}_{5500}, \hat{\lambda}^{(2)}_{5500} \ldots, \hat{\lambda}^{(1000)}_{5500}\},$$

where, for $i = 1, 2, \ldots, 1000$, $\hat{\lambda}^{(i)}_{5500}$ is the BWA estimate of λ calculated using $\tilde{\mathbf{s}}^1_{5500}$ and $\tilde{\lambda}^{(i)}$.

The four plots depicted in Figure 7.1.1 show the distributions of $\tilde{\lambda}^{1000}$ and $\hat{\lambda}^{1000}_{5500}$ (labelled as 'Random Inputs' and 'BWA estimates' respectively), separated into each of the four parameters of interest which need to be estimated (namely p_{11}, p_{21}, b_{11} and b_{21}). The true parameter value is depicted in the plots as a bar.

As expected, $\tilde{\lambda}^{1000}$ is uniformly distributed for each of the four parameters. The distribution of $\hat{\lambda}^{1000}_{5500}$ for each of the four parameters appears to be multi-modal (in particular there appears to be two modes in the BWA estimates for p_{11} and p_{21}, and three modes in the BWA estimates for b_{11} and b_{21}). While the BWA seems to estimate the true parameter value well for some of the inputs in $\tilde{\lambda}^{1000}$, in other instances it appears as if the BWA is converging to a local maxima which does not represent

the true parameter value. This may indeed be problematic in practice as typically the true parameter value is unknown and hence it will be unknown if the BWA has converged to the true parameter value or some incorrect value. In addition to this, in some cases the BWA estimate for p_{11} and p_{21} has not converged to one of the modes.

At first glance these results may appear quite discouraging, however further analysis reveals some interesting insights and shows that the BWA is indeed performing well for this particular simulation. To this end scatter plots of $\tilde{\lambda}^{1000}$ and $\hat{\lambda}^{1000}_{5500}$ were graphed and are displayed in Figure 7.1.2. This allows the interaction between the BWA estimates for the four parameters to be analysed. Once again the following can clearly be seen: the uniform randomness in the initial estimates $\tilde{\lambda}^{1000}$ which are used to start training the BWA; the two modes in the BWA estimates for p_{11} and p_{21}; the fact that not all of the BWA estimates for p_{11} and p_{21} have converged to one of the modes; and the three modes in the BWA estimates for b_{11} and b_{21}. Furthermore, it can also be seen that the two apparent modes for the p_{11} and p_{21} BWA estimates coincide and the three apparent modes for the b_{11} and b_{21} BWA estimates also coincide.

In order to gain further insights, separate scatter plots were produced for each of the three apparent modes for b_{11} and b_{21}. These are shown in Figures 7.1.3 (i)-(iii). Analysing these plots lead to the following conclusions:

- If the starting values used to train the BWA ($\tilde{\lambda}^{1000}$) were such that they were situated in the region of the lower right triangle for P (that is $p_{11} > p_{21}$) and the lower right triangle for B (that is $b_{11} > b_{21}$), then the final BWA estimates $\hat{\lambda}^{1000}_{5500}$ tended to be estimated in the vicinity of a mode centred at $p_{11} = 0.725$, $p_{21} = 0.15$, $b_{11} = 0.675$ and $b_{21} = 0.1$. These estimates are close to the true model parameter values.

- If the starting values used to train the BWA ($\tilde{\lambda}^{1000}$) were such that they were situated in the region of the lower right triangle for P (that is $p_{11} > p_{21}$) and the upper left triangle for B (that is $b_{11} < b_{21}$), then the final BWA estimates

$\hat{\lambda}_{5500}^{1000}$ tended to be estimated in the vicinity of a mode centred at $p_{11} = 0.85$, $p_{21} = 0.275$, $b_{11} = 0.1$ and $b_{21} = 0.675$. These estimates are further discussed below.

- If the starting values used to train the BWA ($\tilde{\lambda}^{1000}$) were such that they were situated in the region of the upper left triangle for P (that is $p_{11} < p_{21}$) then the BWA estimates for b_{11} and b_{21} tended to be estimated in the vicinity of a mode $b_{11} = 0.3$ and $b_{21} = 0.3$, while BWA estimates for p_{11} and p_{21} did not significantly change from their starting BWA estimates. These estimates are further discussed below.

It should at this point be noted that a HMM with parameters $p_{11} = 0.85$, $p_{21} = 0.3$, $b_{11} = 0.1$, $b_{21} = 0.7$ is equivalent in terms of statistical properties to the HMM with parameters specified in λ (all that has changed is that the state labels have been permuted). Furthermore, the BWA parameter estimates mentioned in the second point above align closely with these parameter values. This then explains the symmetry which can be viewed in the BWA estimates around the lines $p_{21} = 1 - p_{11}$ and $b_{21} = b_{11}$ in the scatter plots of Figure 7.1.2. In order to make the BWA estimates $\hat{\lambda}_{5500}^{1000}$ more identifiable, all estimates for (p_{11}, p_{21}) which lie above the diagonal $p_{21} = 1 - p_{11}$ were reflected around $p_{21} = 1 - p_{11}$. The associated (b_{11}, b_{21}) estimates for these points were reflected around the diagonal $b_{21} = b_{11}$. This gives rise to the scatter plot in Figure 7.1.4. It can be observed that once reflected, the BWA estimates originally clustered around $(p_{11}, p_{21}) = (0.85, 0.275)$ and $(b_{11}, b_{21}) = (0.1, 0.675)$ are now transformed to estimates clustered around $(p_{11}, p_{21}) = (0.725, 0.15)$ and $(b_{11}, b_{21}) = (0.675, 0.1)$, which of course closely resemble the true parameter values of λ.

The cluster of BWA estimates around the point $(b_{11}, b_{21}) = (0.3, 0.3)$ in Figure 7.1.4 appear to correspond to a process where the signal distribution for each state follows a Bernoulli process with probability 0.3. To see this note that since for this cluster the rows of $\hat{B}$ are effectively equivalent, the state is irrelevant to the signal which is

emitted (i.e., the signal emitted will be 1 with probability 0.3 and 2 with probability 0.7 for either state). As mentioned earlier in this section, the marginal probability for the HMM λ outputting signal 1 is 0.3. This then explains the cluster of BWA estimates around the point $(b_{11}, b_{21}) = (0.3, 0.3)$ and also why, for these points, the associated BWA estimates for (p_{11}, p_{21}) are scattered - see Figure 7.1.3 (iii) (since the state occupied has no influence on the signal emitted, the state transition probabilities can be any value without effecting the outputted signal sequence).

Next the likelihood value for each of the BWA parameter set estimates within $\hat{\lambda}_{5500}^{1000}$ was calculated; that is for each $i = 1, 2, \ldots, 1000$ the following was calculated:

$$l_{5500}^{(i)} = P(\mathbf{S}_{5500} = \tilde{\mathbf{s}}_{5500}^{1} | \hat{\lambda}_{5500}^{(i)}) \, .$$

The BWA estimate corresponding to the maximum of $\{l_{5500}^{(1)}, l_{5500}^{(2)}, \ldots, l_{5500}^{(1000)}\}$ is $(\hat{p_{11}}, \hat{p_{21}})$ $= (0.87, 0.27)$ and $(\hat{b_{11}}, \hat{b_{21}}) = (0.11, 0.67)$. Once reflected, this yields $(\hat{p_{11}}, \hat{p_{21}}) = (0.73, 0.13)$ and $(\hat{b_{11}}, \hat{b_{21}}) = (0.67, 0.11)$. These estimates closely resemble the true parameter values $(p_{11}, p_{21}) = (0.70, 0.15)$ and $(b_{11}, b_{21}) = (0.70, 0.10)$. This is encouraging especially if one considers that no data from the hidden state sequence forms part of the training data for the BWA.

From the above analysis key observations can be made. Firstly the starting value for the BWA may play a significant role in determining what the final BWA estimate will be. For this reason it is advised that several different starting values be used as was the case in the above exercise. Secondly, once the BWA has been performed and a HMM has been fit, one should take care in labelling and interpreting the states. To this end appropriate reflection of the BWA estimates may be required as was considered in the analysis above. Finally it can also be seen that if interpreted correctly, the BWA may indeed be effective in estimating the true model parameters for the HMM.

Of course it should be noted that a signal sequence of length 5500 was used to train the BWA in the above exercise. The next question of interest is how the results of

the above exercise compare when a shorter signal sequence is used. To this end signal sequences of length 1000, 500, 250, 150 and 75 were simulated, denoted as $\tilde{\mathbf{s}}^1_{1000}$, $\tilde{\mathbf{s}}^1_{500}$, $\tilde{\mathbf{s}}^1_{250}$, $\tilde{\mathbf{s}}^1_{150}$ and $\tilde{\mathbf{s}}^1_{75}$ respectively, and the above exercise was repeated for each simulated signal sequence. Figure 7.1.5 shows both the final BWA estimates and reflected final BWA estimates for (p_{11}, p_{21}) and (b_{11}, b_{21}) when $\tilde{\mathbf{s}}^1_{1000}$ was used to train the BWA. The graphs depicting the corresponding results when $\tilde{\mathbf{s}}^1_{500}$, $\tilde{\mathbf{s}}^1_{250}$, $\tilde{\mathbf{s}}^1_{150}$ and $\tilde{\mathbf{s}}^1_{75}$ were used to train the BWA are shown in Figures 7.1.6 to 7.1.9.

Next the BWA estimates corresponding to the maximum of $\{l^{(1)}_{1000}, l^{(2)}_{1000}, \ldots, l^{(1000)}_{1000}\}$, $\{l^{(1)}_{500}, l^{(2)}_{500}, \ldots, l^{(1000)}_{500}\}$, $\{l^{(1)}_{250}, l^{(2)}_{250}, \ldots, l^{(1000)}_{250}\}$, $\{l^{(1)}_{150}, l^{(2)}_{150}, \ldots, l^{(1000)}_{150}\}$ and $\{l^{(1)}_{75}, l^{(2)}_{75}, \ldots, l^{(1000)}_{75}\}$ were determined and reflected where necessary. These parameter estimates are summarised in the table below (together with the results given earlier when $\tilde{\mathbf{s}}^1_{5500}$ was used to train the BWA).

	$\hat{p}_{11}$	$\hat{p}_{21}$	$\hat{b}_{11}$	$\hat{b}_{21}$
True parameter value	**0.70**	**0.15**	**0.70**	**0.10**
BWA estimate from max of $\{l^{(1)}_{5500}, l^{(2)}_{5500}, \ldots, l^{(1000)}_{5500}\}$	0.73	0.13	0.67	0.11
BWA estimate from max of $\{l^{(1)}_{1000}, l^{(2)}_{1000}, \ldots, l^{(1000)}_{1000}\}$	0.68	0.24	0.58	0.05
BWA estimate from max of $\{l^{(1)}_{500}, l^{(2)}_{500}, \ldots, l^{(1000)}_{500}\}$	0.73	0.17	0.70	0.03
BWA estimate from max of $\{l^{(1)}_{250}, l^{(2)}_{250}, \ldots, l^{(1000)}_{250}\}$	0.90	0.02	0.67	0.15
BWA estimate from max of $\{l^{(1)}_{150}, l^{(2)}_{150}, \ldots, l^{(1000)}_{150}\}$	0.48	0.24	0.99	0.00
BWA estimate from max of $\{l^{(1)}_{75}, l^{(2)}_{75}, \ldots, l^{(1000)}_{75}\}$	0.39	0.26	0.99	0.02

Table 7.1.1: BWA estimates using simulated signal sequences of different lengths

Figures 7.1.4 - 7.1.9 and the above table show, as expected, that the performance of the BWA deteriorates as shorter data sequences are used to train the algorithm. Considering that the chosen λ is a HMM of the most basic form (i.e. five unknown parameters), the poor performance of the BWA when a signal sequence of length 75 or 150 is used for training suggests that the BWA is quite a 'data hungry' algorithm. This may be attributed to the fact that no direct data from the hidden state process is available to train the model parameters. The results do however show that if enough data is available for training, the BWA can be effective in determining the true model

parameters of the HMM.

The stability of the above simulation exercise was also tested. For each signal sequence $\tilde{\mathbf{s}}_k^1$, where $k \in \{75, 150, 250, 500, 1000, 5500\}$, the BWA estimates from the 50 largest values of $\{l_k^{(1)}, l_k^{(2)}, \ldots, l_k^{(1000)}\}$ (as opposed to just the maximum) were compared. Each of these estimates gave similar results to the corresponding estimate in Table 7.1.1.

Recall that for a given signal length in above simulations, the same signal sequence was used for each of the 1000 iterations. That is for $k \in \{75, 150, 250, 500, 1000, 5500\}$ the same signal sequence $\tilde{\mathbf{s}}_k^1$ was used to train the BWA and obtain estimates $\hat{\lambda}_k^{1000} = \{\hat{\lambda}_k^{(1)}, \hat{\lambda}_k^{(2)} \ldots, \hat{\lambda}_k^{(1000)}\}$. Only the starting values for the BWA changed. Of course the simulated signal sequence used to train the BWA estimates could also have an influence on the final BWA estimates. For example, a second state and signal sequence of length 75 was simulated and the above simulation exercise repeated. For this simulated signal sequence, the BWA estimates from the maximum of $\{l_{75}^{(1)}, l_{75}^{(2)}, \ldots, l_{75}^{(1000)}\}$ was $(\hat{p_{11}}, \hat{p_{21}}) = (0.83, 0.03)$ and $(\hat{b_{11}}, \hat{b_{21}}) = (0.85, 0.16)$. This is a material improvement to the BWA estimates given in Table 7.1.1. The corresponding graph of Figure 7.1.9 also shows that the BWA is producing significantly improved estimates (when compared to the actual parameter values) for this particular simulated signal sequence. This suggests that a study of the sampling distribution of the BWA estimates given in Table 7.1.1 would prove insightful. This is examined in more detail in the next section.

To summarise the above discussion, there appears to be three significant drivers of the accuracy of BWA, namely (i) the starting values used for the BWA, (ii) the length of the signal sequence used to train the BWA, and (iii) the actual data sequence used to train the BWA. Points (i) and (ii) have been detailed in the preceding discussion. The next section explores point (iii) and also adds further discussion to point (ii).

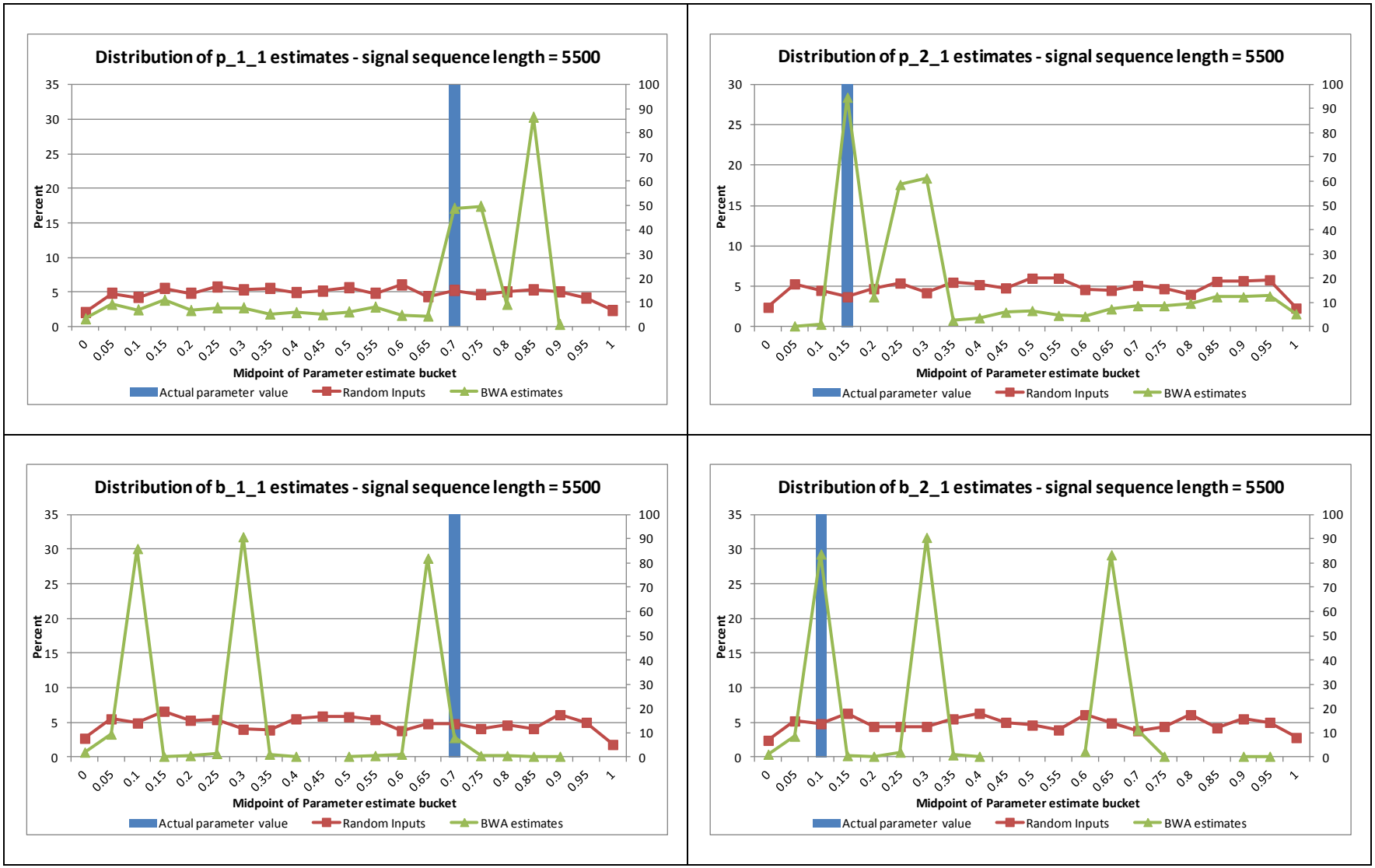


Figure 7.1.1: Frequency curves of BWA estimates using a signal sequence of 5500

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

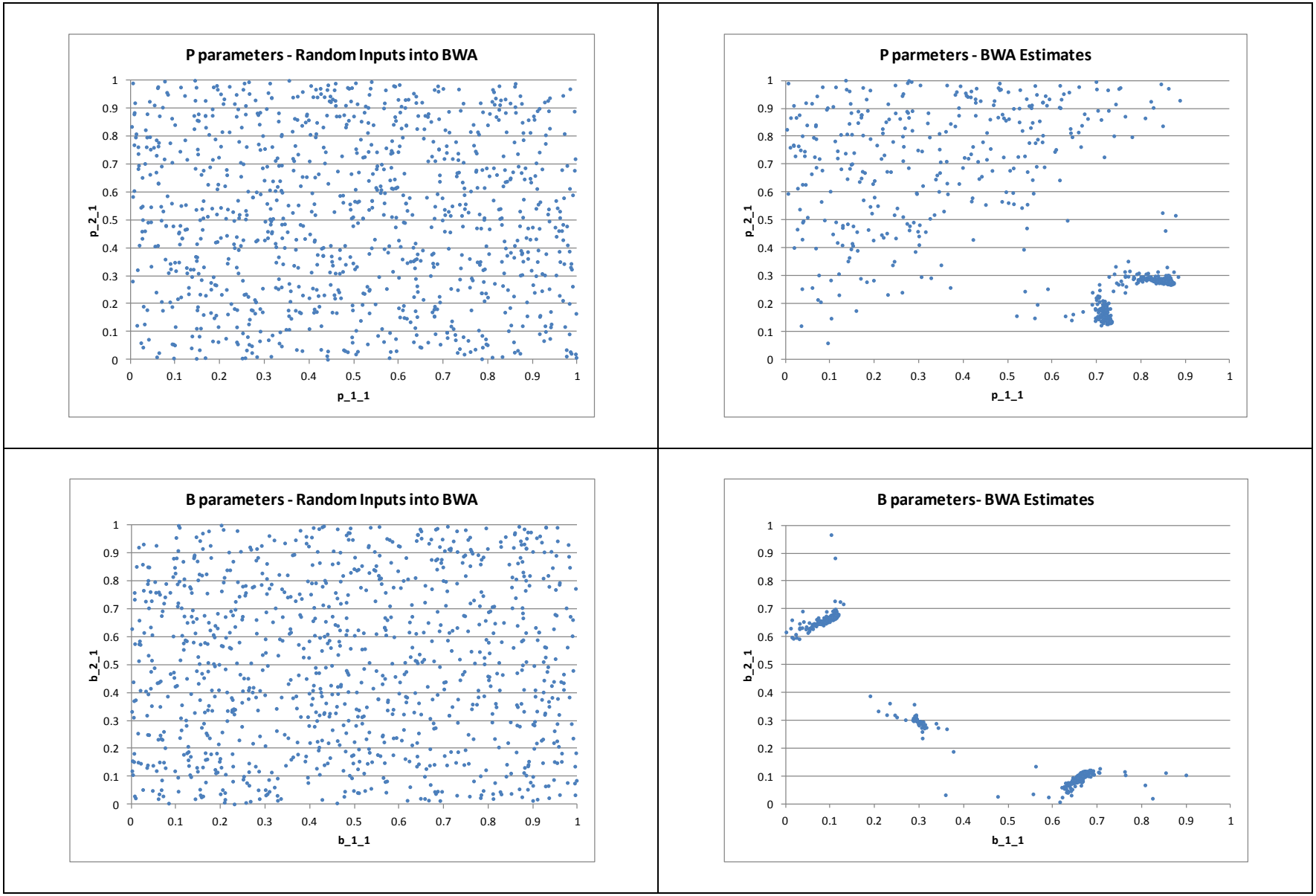


Figure 7.1.2: Scatter plot of BWA estimates using a signal sequence of 5500

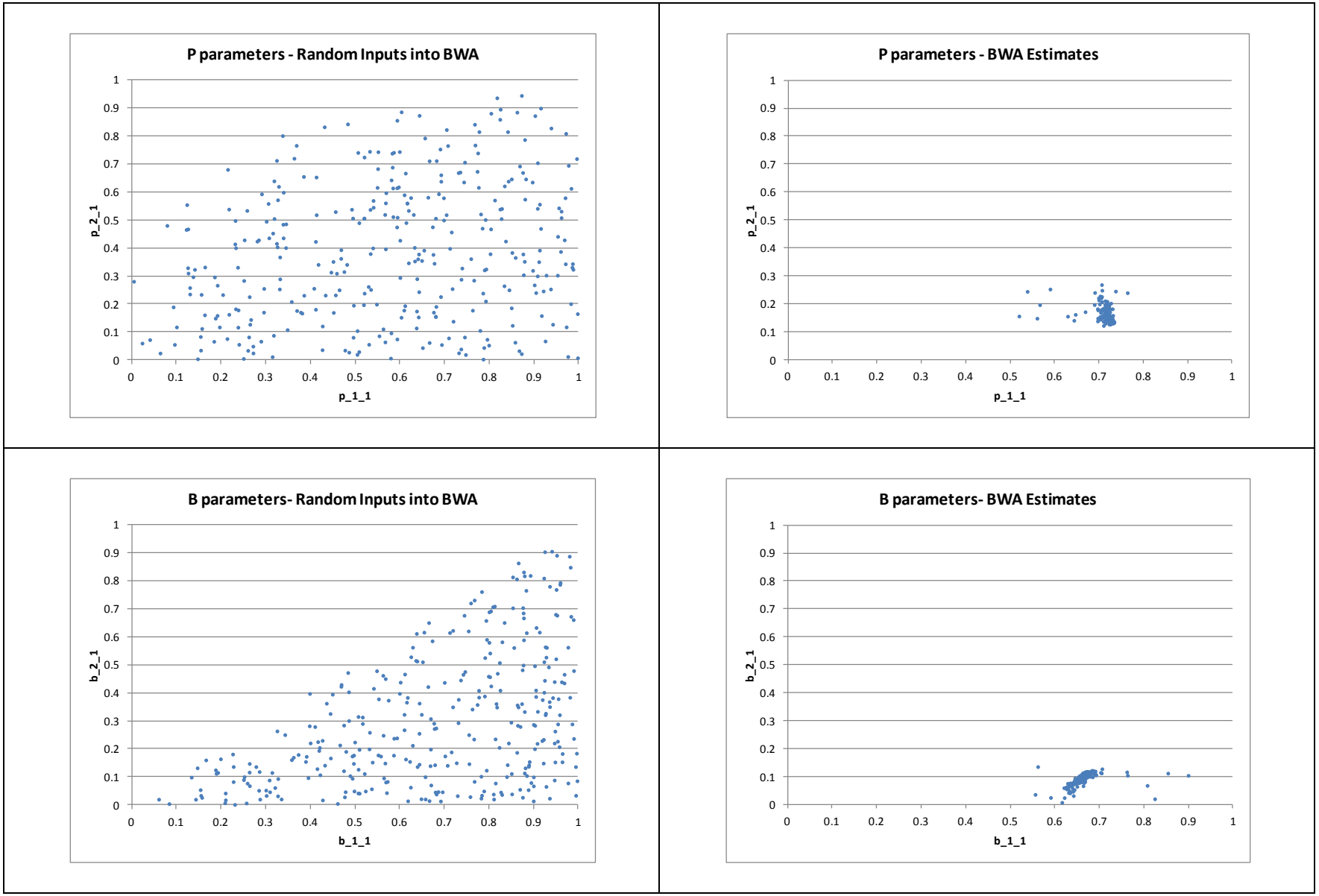


Figure 7.1.3 (i): Mode 1 for the B BWA estimates using a signal sequence of 5500

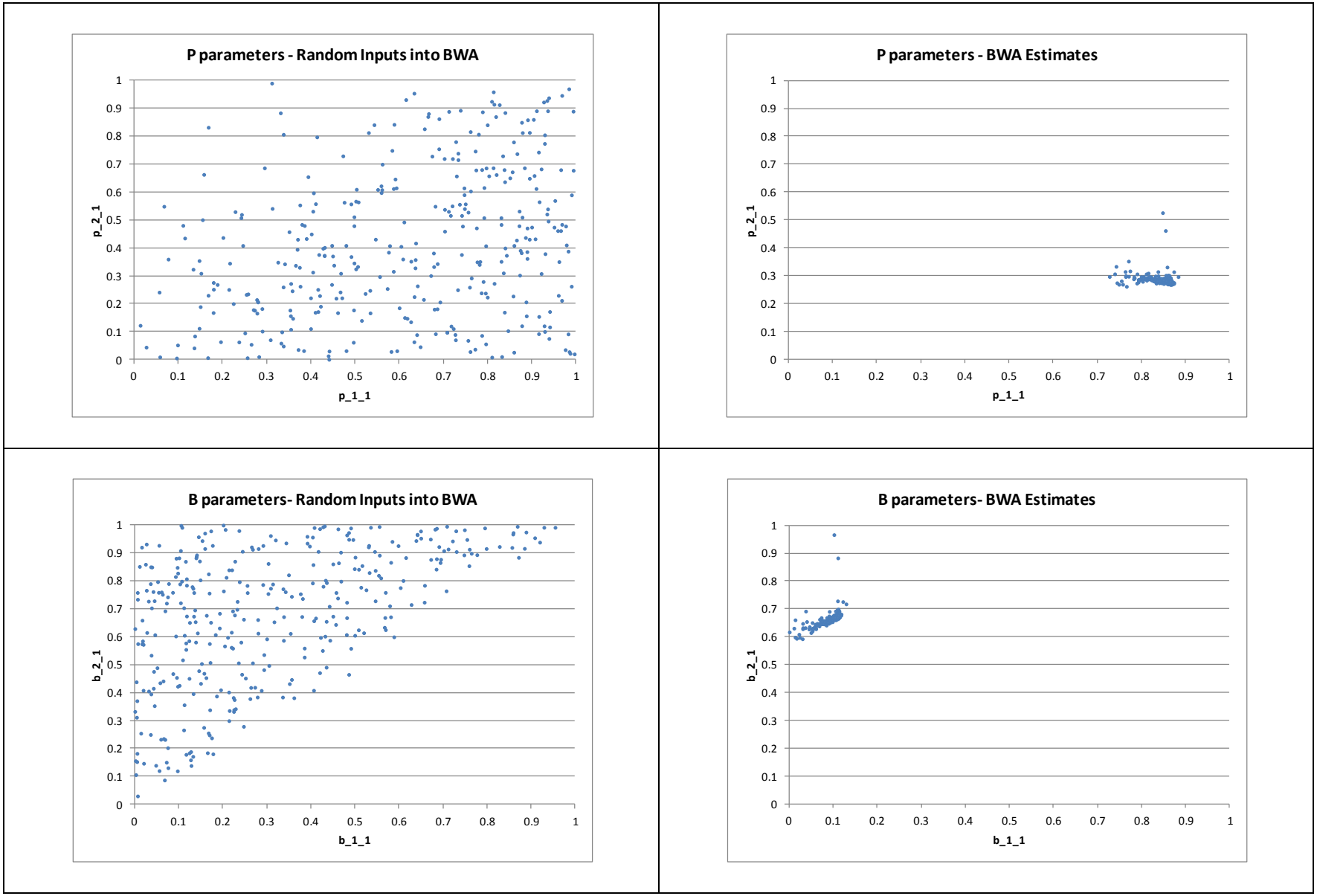


Figure 7.1.3 (ii): Mode 2 for the B BWA estimates using a signal sequence of 5500

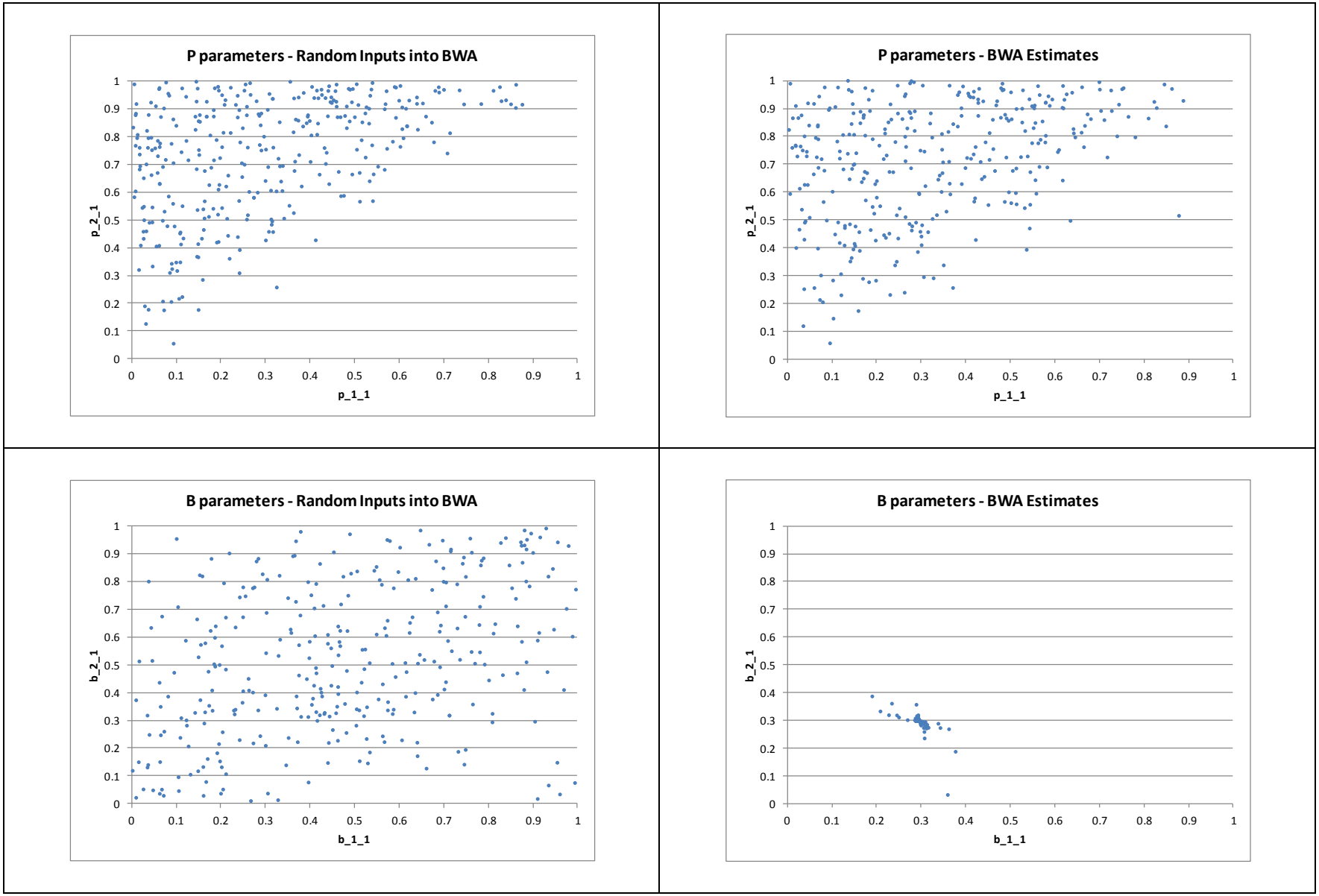


Figure 7.1.3 (iii): Mode 3 for the B BWA estimates using a signal sequence of 5500

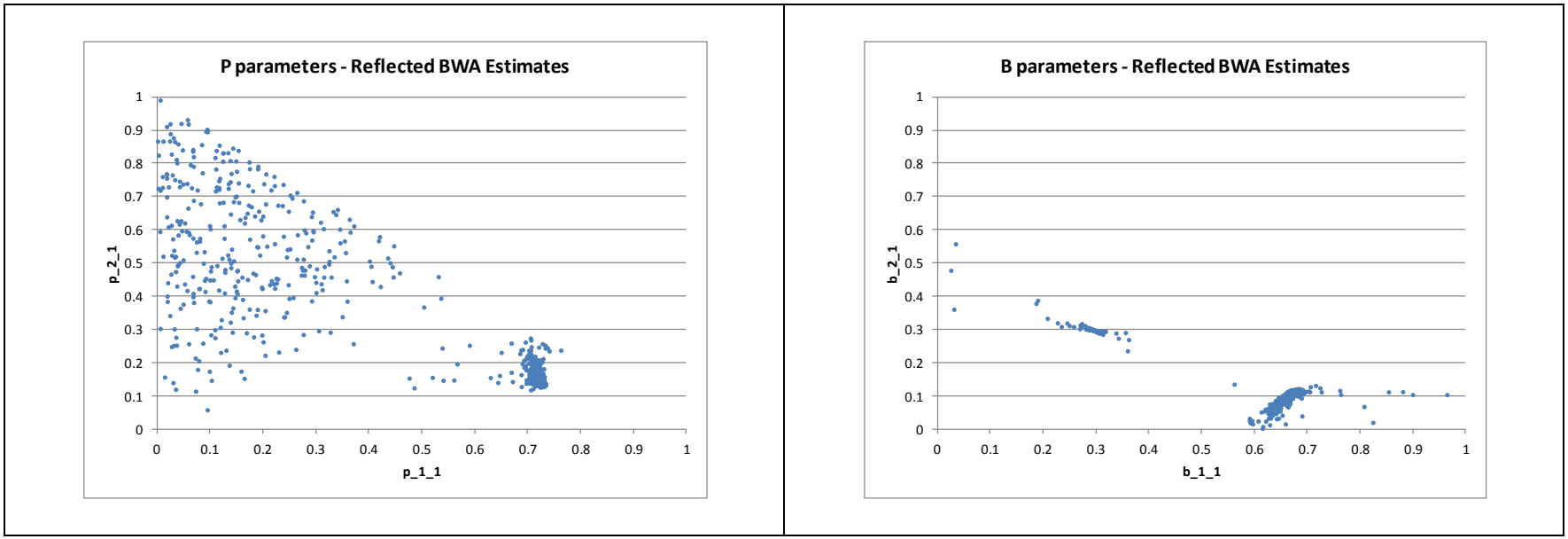


Figure 7.1.4: Reflected Scatter plot of BWA estimates using a signal sequence of 5500

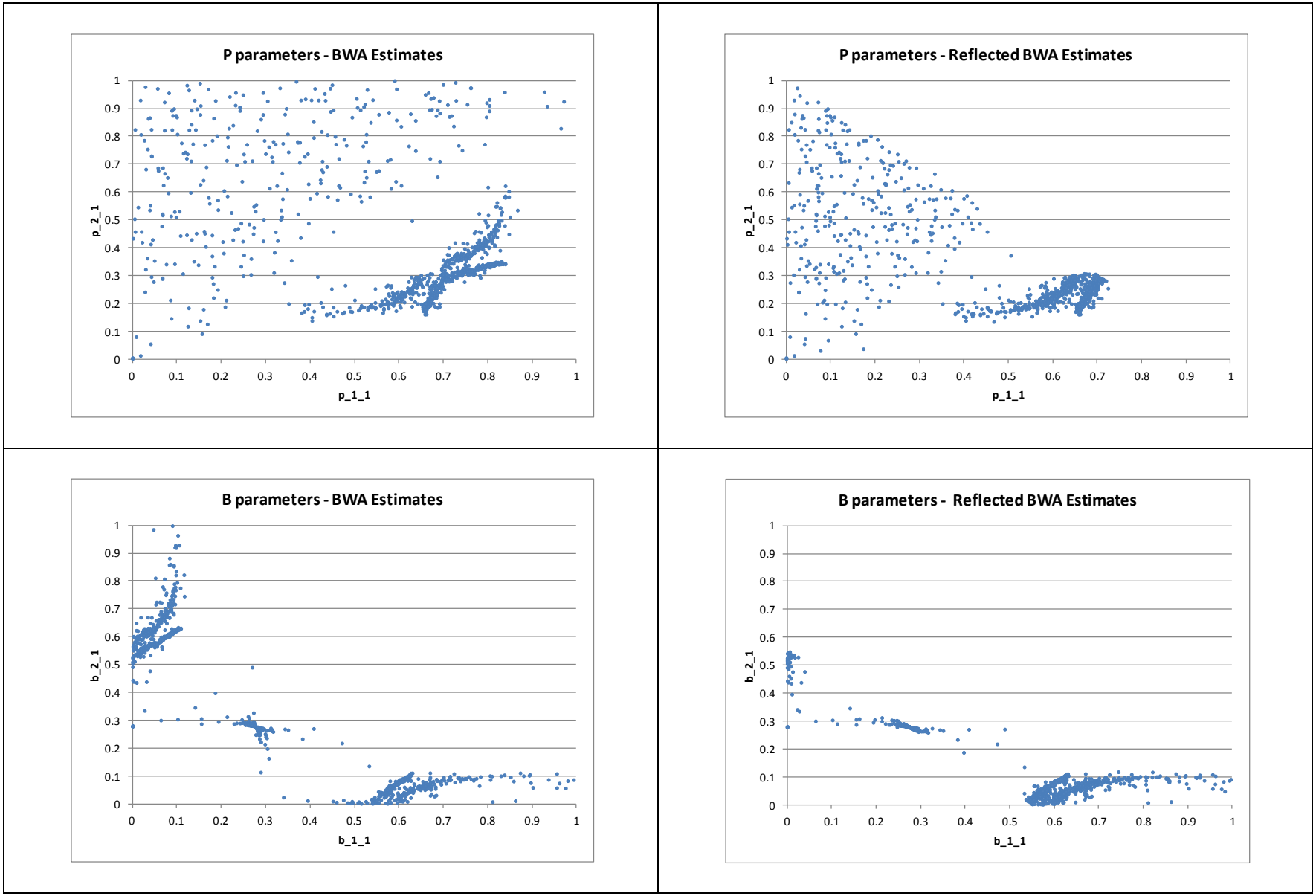


Figure 7.1.5: Scatter plot of BWA estimates using a signal sequence of 1000

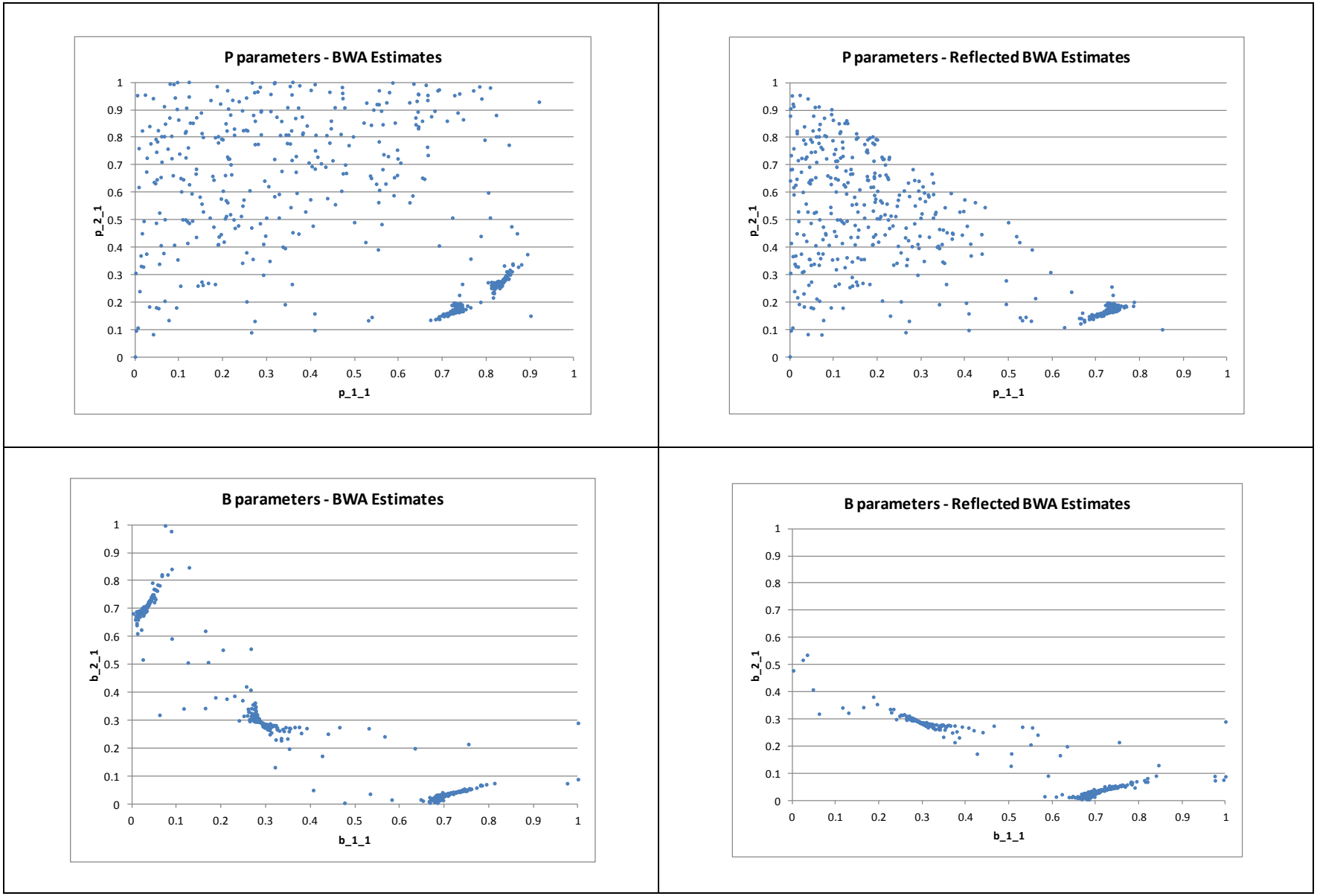


Figure 7.1.6: Scatter plot of BWA estimates using a signal sequence of 500

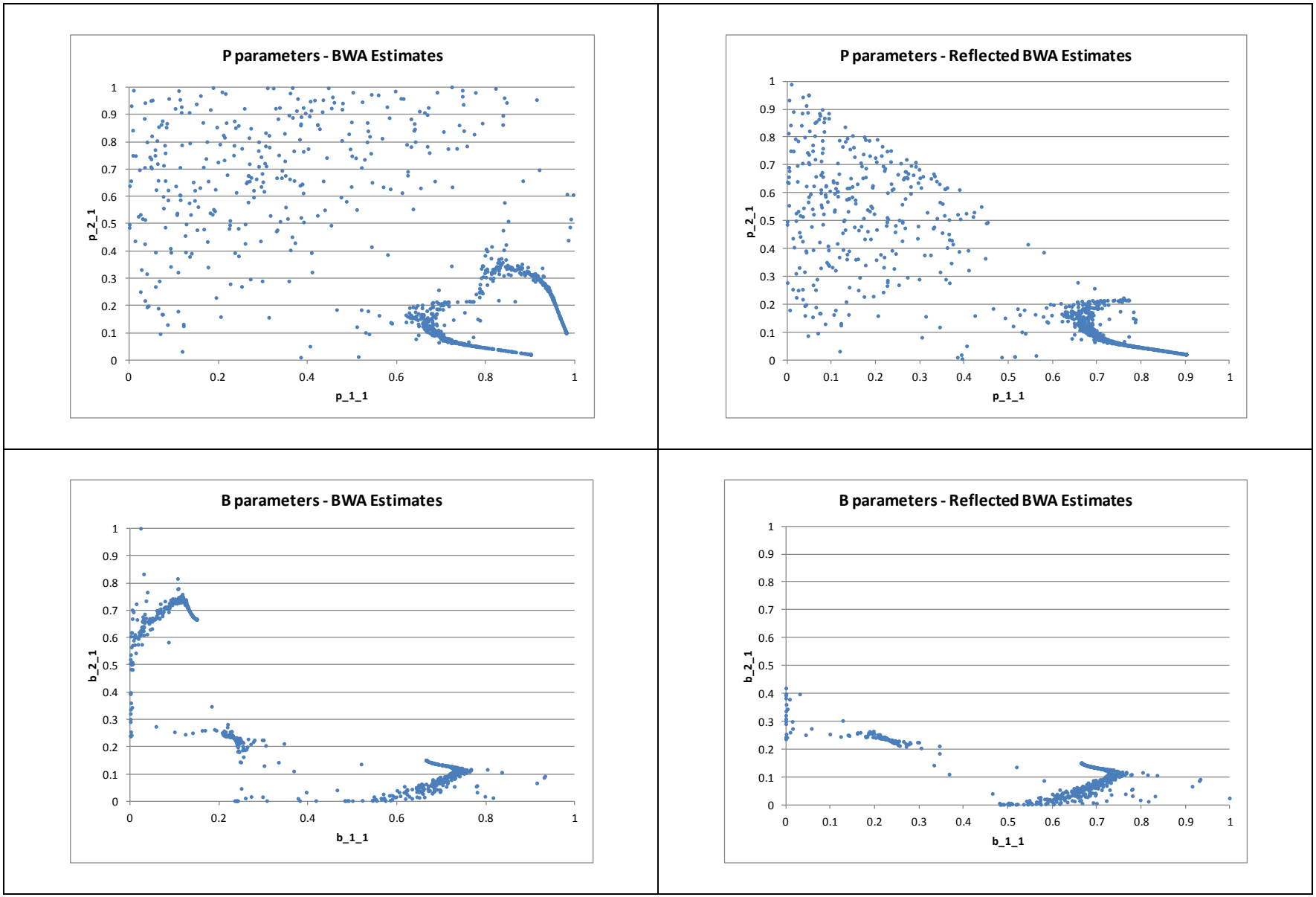


Figure 7.1.7: Scatter plot of BWA estimates using a signal sequence of 250

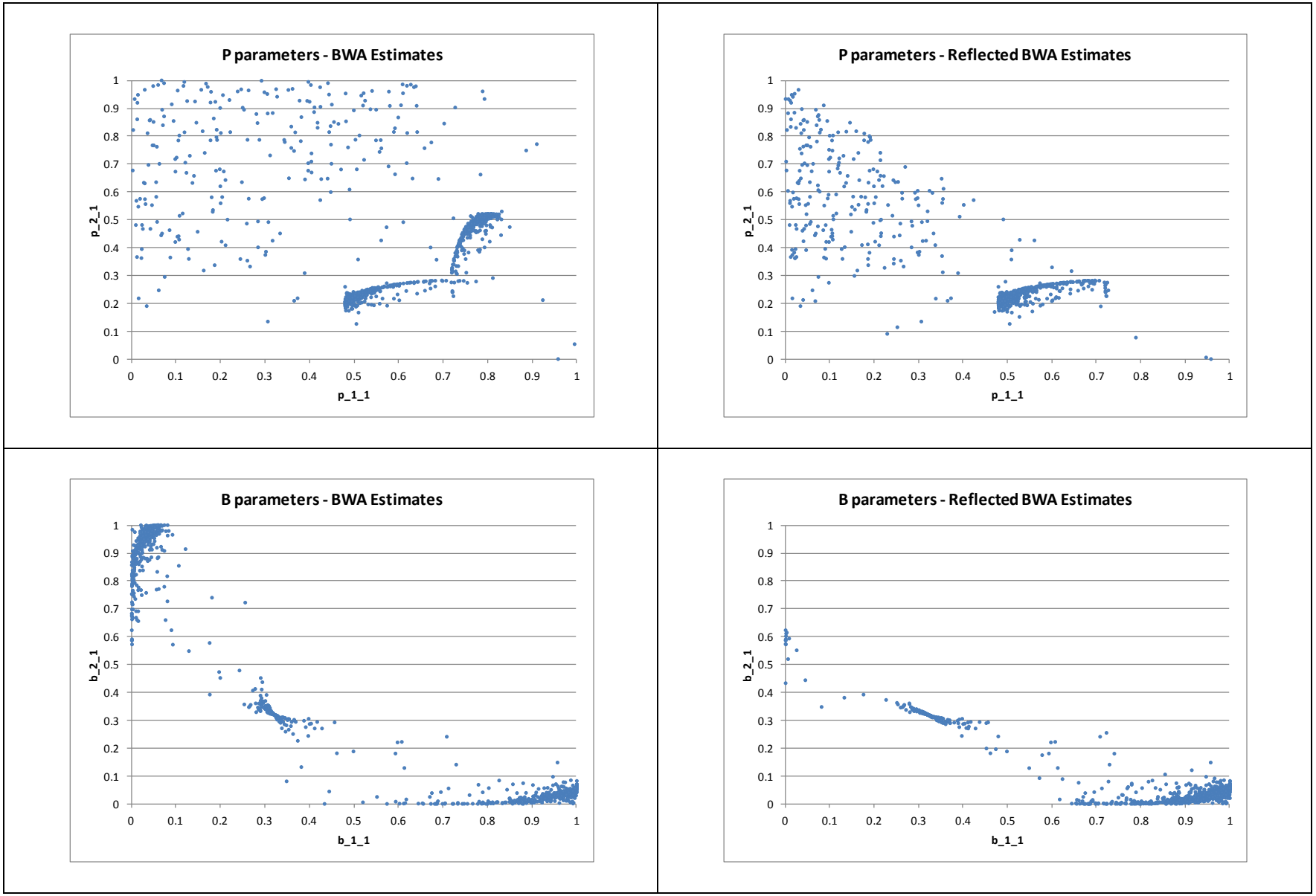


Figure 7.1.8: Scatter plot of BWA estimates using a signal sequence of 150

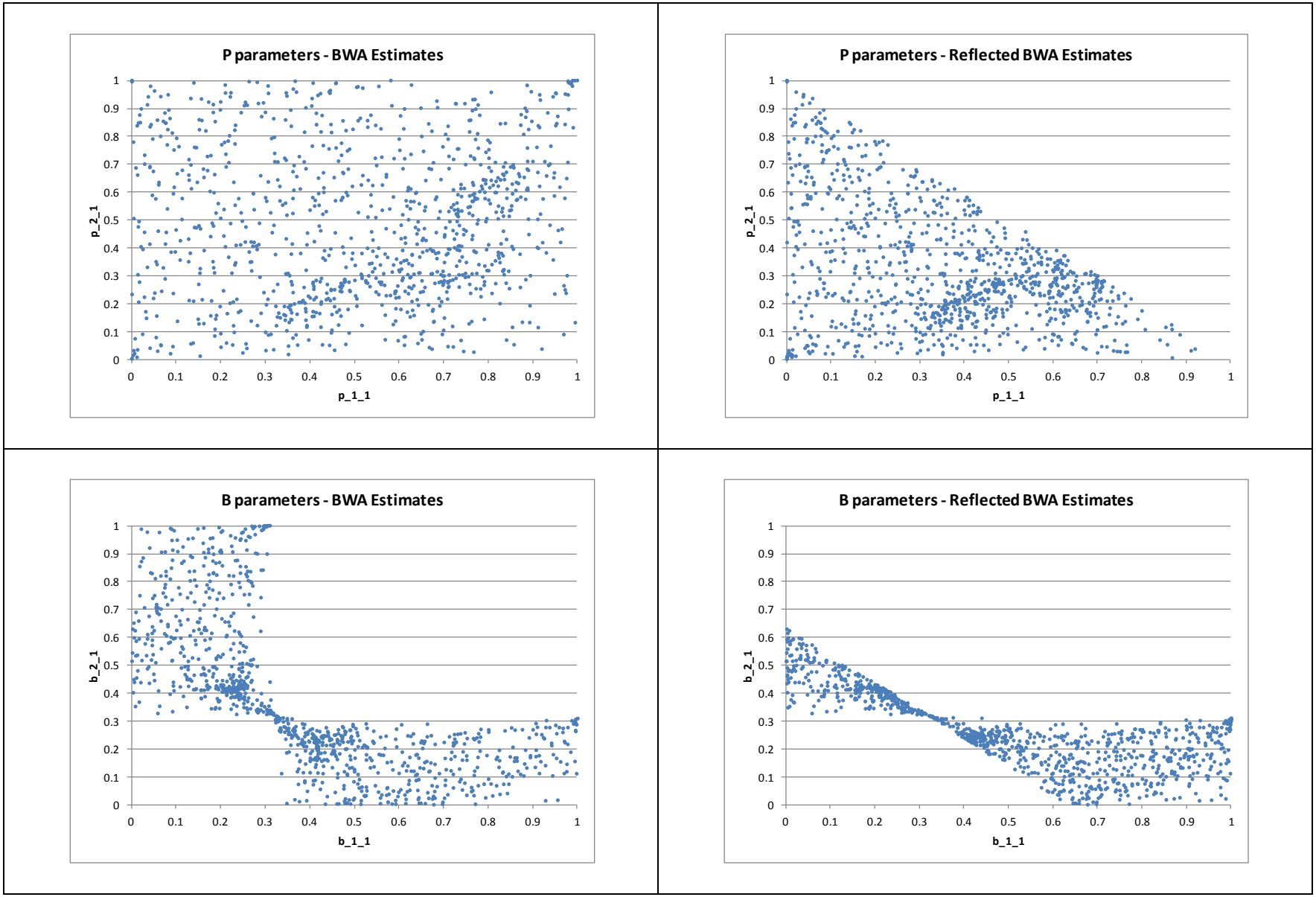


Figure 7.1.9: Scatter plot of BWA estimates using a signal sequence of 75

7.1.2 Investigating Sampling Distributions of Baum-Welch Algorithm Estimates

The above section discussed the influence which different starting values and signal sequence lengths have on the BWA estimates. In particular, for a given signal sequence, the BWA was trained using different starting values (randomly assigned using the uniform distribution) and the BWA estimate corresponding to the maximum likelihood value was determined. This was performed using different signal sequence lengths, resulting in the BWA estimates shown in Table 7.1.1. It should be noted that for each signal sequence length, the signal sequence used for training was kept constant. This section enhances the previous section by now investigating the sampling distributions of the BWA estimates from Table 7.1.1 (that is using various simulated signal sequences to train the BWA rather than a constant signal sequence as was the case in the above section). This is performed for selected signal sequence lengths.

To begin the structure of the simulation exercise is described. Assume the same HMM $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$ which was specified above. Using λ, 500 distinct state and signal sequences were simulated, each of length 250. Let these signal sequences be denoted by

$$\tilde{\mathbf{s}}_{250}^{1}, \tilde{\mathbf{s}}_{250}^{2}, \ldots, \tilde{\mathbf{s}}_{250}^{500}.$$

Now for each simulated signal sequence $\tilde{\mathbf{s}}_{250}^{k}$, where $k \in \{1, 2, \ldots 500\}$, 150 distinct starting values for the BWA were randomly created from the uniform(0,1) distribution (in the same manner which was described above). Let these be denoted by

$$\tilde{\lambda}_{k}^{150} = \{\tilde{\lambda}_{k}^{(1)}, \tilde{\lambda}_{k}^{(2)} \ldots, \tilde{\lambda}_{k}^{(150)}\}.$$

In other words, for each $k = 1, 2, \ldots, 500$ a signal sequence of length of 250, $\tilde{\mathbf{s}}_{250}^{k}$, is simulated and the BWA is performed 150 times, using the 150 randomly created distinct starting values contained in $\tilde{\lambda}_{k}^{150}$. Thus for each k, 150 distinct BWA estimates

are calculated, denoted by

$$\hat{\lambda}_k^{150} = \{\hat{\lambda}_k^{(1)}, \hat{\lambda}_k^{(2)} \ldots, \hat{\lambda}_k^{(150)}\}\,.$$

The likelihood value for each of the BWA parameter set estimates within $\hat{\lambda}_k^{150}$ is then calculated; that is for each $i = 1, 2, \ldots, 150$ the following is calculated:

$$l_k^{(i)} = P(\mathbf{S}_{250} = \tilde{\mathbf{s}}_{250}^k | \hat{\lambda}_k^{(i)})\,.$$

Let the BWA estimate corresponding to the maximum of $\{l_k^{(1)}, l_k^{(2)}, \ldots, l_k^{(150)}\}$ be denoted by λ_k^*. That is

$$\lambda_k^* = \arg\max_{\hat{\lambda}_k^{(i)}} \{l_k^{(i)} \,:\, i = 1, 2, \ldots, 150\}\,.$$

This process is repeated for each k, yielding 500 final BWA estimates calculated using 500 distinct simulated signal sequences and 150 distinct starting inputs into the BWA for each of the 500 simulated signal sequences. Let this be denoted as

$$\lambda^* = \{\lambda_1^*, \lambda_2^*, \ldots, \lambda_{500}^*\}.$$

Thus, λ^* is a (simulated) set of realisations from the sampling distribution of $\hat{\lambda}$ when the sequence length is 250. Assessment of the distribution of λ^* gives light into the sensitivity of the BWA to the signal sequence used to train it. Importantly assessment of the distribution of λ^* will also give insight into the statistical properties (such as bias and variability) of the estimates described in Table 7.1.1. This analysis may prove to be valuable if the technique used to derive Table 7.1.1 is used in practice to recover the true HMM parameters.

The results presented in Figures 7.1.10 and 7.1.11 are the 500 estimates contained in λ^* which have been reflected where appropriate according to the estimates for b_{11} and b_{21}. For a given $i \in \{1, 2, \ldots, 500\}$, this reflection is such that if the (b_{11}, b_{21}) estimates from λ_i^* lie above the diagonal $b_{21} = b_{11}$ then the estimates for (b_{11}, b_{21}) are reflected around the diagonal $b_{21} = b_{11}$. The associated (p_{11}, p_{21}) estimates for these

points are reflected around the diagonal $p_{21} = 1 - p_{11}$.

Frequency distributions of the 500 parameter estimates for p_{11}, p_{21}, b_{11} and b_{21} are shown in Figure 7.1.10. This is followed by Figure 7.1.11 which depicts the interaction between the 500 BWA estimates through scatter plots. Also included in Figure 7.1.11 are scatter plots of the 500 parameter estimates for p_{11}, p_{21}, b_{11} and b_{21} before the reflection is performed. From this the symmetry in the parameter estimates can clearly be seen, especially in the estimates for b_{11} and b_{21} - hence justification for the reflection applied.

The mean squared error (MSE) for λ^* is also examined. Recall that for some parameter y with estimator values $\hat{y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$, the MSE is given by

$$\begin{aligned} MSE(\hat{y}) &= \frac{1}{n}\sum_{i=1}^{n}[\hat{y}_i - y]^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}[\hat{y}_i - \text{mean}(\hat{y}) + \text{mean}(\hat{y}) - y]^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}[\hat{y}_i - \text{mean}(\hat{y})]^2 + [y - \text{mean}(\hat{y})]^2 + 0 \\ &= \text{Var}(\hat{y}) + \text{Bias}(\hat{y}, y)^2 \,. \end{aligned}$$

The MSE was assessed for each of the four parameters of interest in λ^*. The results are shown in the table below.

Parameter	MSE(λ^*)	mean(λ^*)	Bias(λ^*, λ)^2	Var(λ^*)
p_{11}	0.0305	0.7007	0.0000	0.0305
p_{21}	0.0120	0.1511	0.0000	0.0120
b_{11}	0.0417	0.7484	0.0023	0.0393
b_{21}	0.0057	0.0870	0.0002	0.0055

Table 7.1.2: Mean squared error analysis of λ^*, using a sequence length of 250

Analysis of Figures 7.1.10 and 7.1.11 and Table 7.1.2 reveal that the p_{11} and p_{21} parameter estimates contained in λ^* are approximately centred around the true parameter values and show very little bias. When assessing the parameter estimates for b_{11} and b_{21} it is noted that a significant portion of the estimates lie on the boundary.

In particular, 17% of the estimates for b_{11} lie in the interval $(0.99, 1]$ and 24% of the estimates for b_{21} lie in the interval $[0, 0.01)$. The percentage of estimates which lie in both these intervals (that is the estimate for b_{11} is in $(0.99, 1]$ and the estimate for b_{21} is in $[0, 0.01)$) is 1.4%. This introduces some degree of bias in the estimates for b_{11} and b_{21}. The occurrence of parameters estimated on the boundary of their parameter space is also noted in [46] (see page 53) where it is stated that "... some of the parameters are on the boundary of their parameter space, which occurs quite often when HMMs are fitted".

The spread of the parameter estimates λ^* is significant, particularly the estimates for the p_{11} and b_{11} parameters. This variability in the parameter estimates λ^* can come from one of two sources. Suppose that $i, j \in \{1, 2, \ldots, 500\}$ such that the estimate λ_i^* is significantly different from the estimate λ_j^*. This difference can either be due to (a) the number (150 in this simulation exercise) of random starting values used for the BWA not being sufficient large enough, or (b) the simulated signal sequence differing between run i and run j of the simulation exercise (that is sampling variability). Analysis revealed that case (a) does not seem to be a significant contributor to the variability in λ^*.[1] This suggests that the variability in λ^* is due to case (b) rather than case (a). That is, the different simulated signal sequence used in run i and run j seems to be the driver in λ_i^* and λ_j^* differing, and hence the sampling variability in λ^*. This highlights the sensitivity of the BWA to the signal sequence used to train the estimates - a somewhat expected result.

Finally, review of Figures 7.1.10 and 7.1.11 highlights the presence of four extreme outliers in the estimates for p_{11} and p_{21}. Analysis which was performed showed that the cause of these outlying estimates was due to the signal sequence used to train the BWA rather than the random inputs used as starting inputs into the BWA. This is consistent with the discussion in the above paragraph. Further analysis also showed

[1] Analysis was performed by using the 150 starting values from simulation run j and the simulated signal sequence from run i to train the BWA. The estimates which were obtained were very comparable to λ_i^*. This was tested for various $i, j \in \{1, 2, \ldots, 500\}$.

that these outlying estimates did not significantly alter the conclusions derived from the MSE results presented in Table 7.1.2.

The above exercise was repeated using a sequence length of 1000. That is λ^* now represents 500 BWA estimates calculated using 500 distinct simulated signal sequences of length 1000 and 150 distinct starting inputs into the BWA for each of the 500 simulated signal sequences. The distribution of this λ^* is summarised in Figures 7.1.12 and 7.1.13 and in the table below.

Parameter	MSE(λ^*)	mean(λ^*)	Bias(λ^*, λ)^2	Var(λ^*)
p_{11}	0.0090	0.6965	0.0000	0.0090
p_{21}	0.0030	0.1557	0.0000	0.0030
b_{11}	0.0152	0.7165	0.0003	0.0149
b_{21}	0.0025	0.0920	0.0001	0.0024

Table 7.1.3: Mean squared error analysis of λ^*, using a sequence length of 1000

These results are compared to the earlier discussed sampling distributions of the BWA estimates when signal sequences of length 250 was used to train the BWA. Once again the parameter estimates are approximately centred around the true parameter values and show very little bias for the p_{11} and p_{21} parameters. Bias is still present in the BWA estimates for b_{11} and b_{21}, however less than was observed in Table 7.1.2. This is largely due to the fact that a notably smaller portion of the estimates lie on the boundary. In particular, 2.8% of the estimates for b_{11} lie in the interval $(0.99, 1]$ and 8.4% of the estimates for b_{21} lie in the interval $[0, 0.01)$. There were no estimates which lie in both these intervals (that is the estimate for b_{11} is in $(0.99, 1]$ and the estimate for b_{21} is in $[0, 0.01)$). The corresponding proportions when a signal length of 250 was used was 17%, 24% and 1.4% respectively.

While the parameter estimates of λ^* do still show spread around the true parameter values, it has decreased when compared to the spread which was observable in the sampling distributions when signal sequences of length 250 was used to train the BWA. This can be seen by comparing Figures 7.1.12 and 7.1.13 to Figures 7.1.10 and 7.1.11; as well as by comparing Table 7.1.3 to Table 7.1.2. In particular, using a

signal sequence length of 1000 led to the variance of the sampling distribution of the parameter estimates decreasing by a factor of 3.4, 4.0, 2.6 and 2.3 for $p_{11}, p_{21}, b_{11}, b_{21}$ respectively (when compared to the variance of the sampling distribution of the parameter estimates when the signal length of 250 was used). This in turn, together with decreased bias, has resulted in decreased mean squared errors (by factors of 3.4, 4.0, 2.7 and 2.3 for $p_{11}, p_{21}, b_{11}, b_{21}$ respectively).

One point to note is that a general rule of thumb is that the variance of the sampling distribution of a parameter estimate should decrease by a factor approximately equal to the factor of the sample size increase. In this particular exercise, the sequence length of each simulation has increased by a factor of 4 (from 250 to 1000), hence the sampling variance of the parameter estimates is expected to decrease by a factor of 4. The variance decrease observed for the p_{11} and p_{21} parameter estimates is in this region (the variance decreases by a factor of 3.4 and 4.0 respectively). However the variance decrease observed for the b_{11} and b_{21} parameter estimates is notably less (the variance decreases by a factor of 2.6 and 2.3 respectively). A search through the literature reveals little to explain this. For example [46] discusses the standard errors of HMM parameter estimates (see Section 3.6) but does not mention the relationship between these standard errors and the sample size. A possible reason for the observed variance decrease for the b_{11} and b_{21} parameter estimates being less than expected could be the number of parameter estimates located on the boundary of the parameter space when a sample size of 250 was used (as was discussed earlier in this section - see Figures 7.10 and 7.11). Had there not been a boundary constraint, these estimates might have been estimated further from their true parameter value and hence the variance would have been larger for a sample size of 250. Thus when the sample size of 1000 is used (and fewer estimates are now located on the boundary), the variance of the parameter estimates for b_{11} and b_{21} decreases by a factor lower than expected.

Encouragingly, no obvious outliers are observable in the sampling distributions when

signal sequences of length 1000 were used to train the BWA.

In summary, increasing the length of the signal sequences used to train the BWA from 250 to 1000 led to following observable properties in the sampling distributions for the parameter estimates: (a) notably less bias in the parameter estimates, (b) notably less spread in the parameter estimates, (c) significantly fewer parameter estimates situated on the boundary for b_{11} and b_{21}, (d) the removal of the extreme outlying parameter estimates.

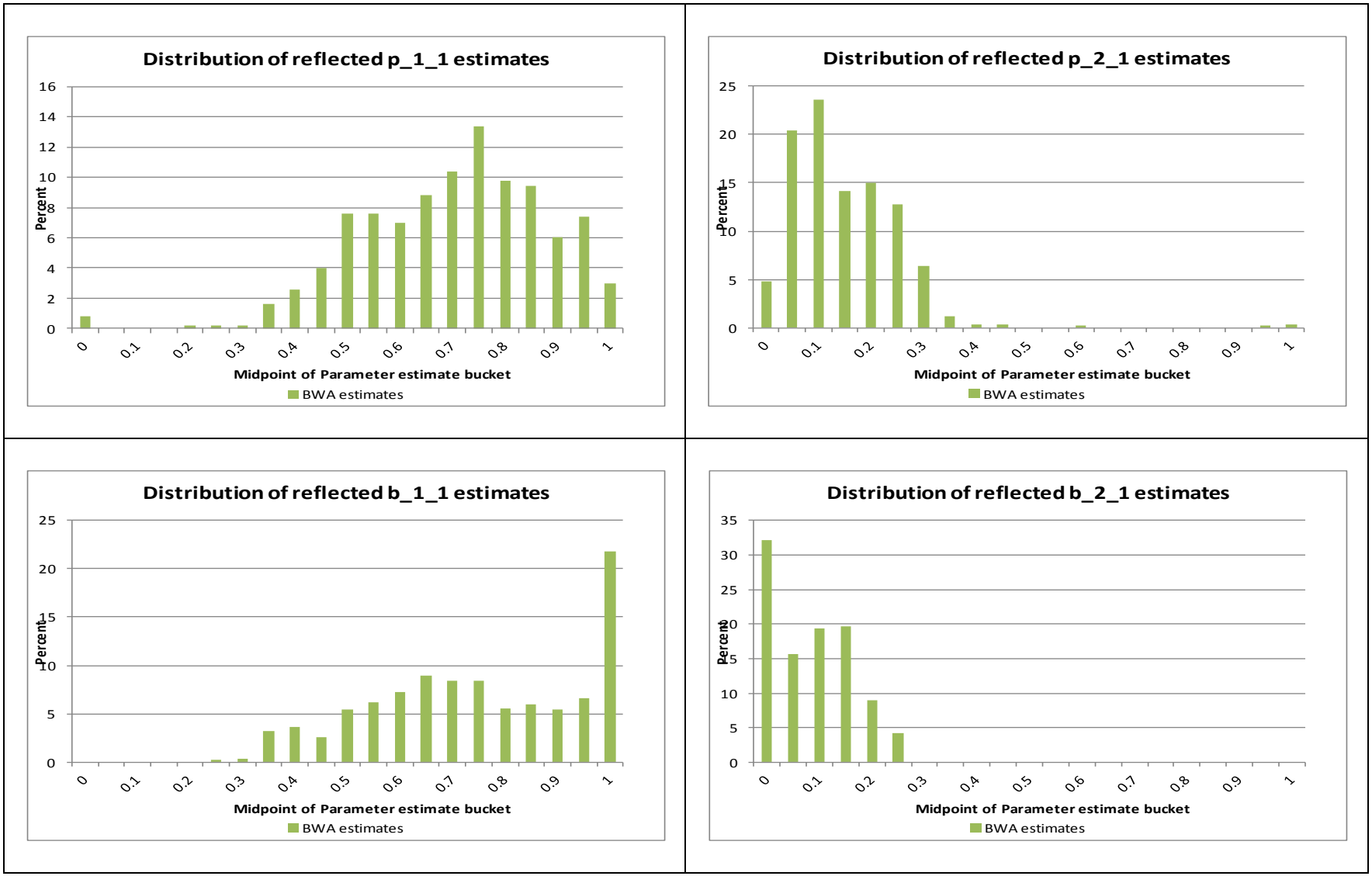


Figure 7.1.10: Frequency curves of the 500 BWA estimates contained in λ^* - using a signal sequence length of 250

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

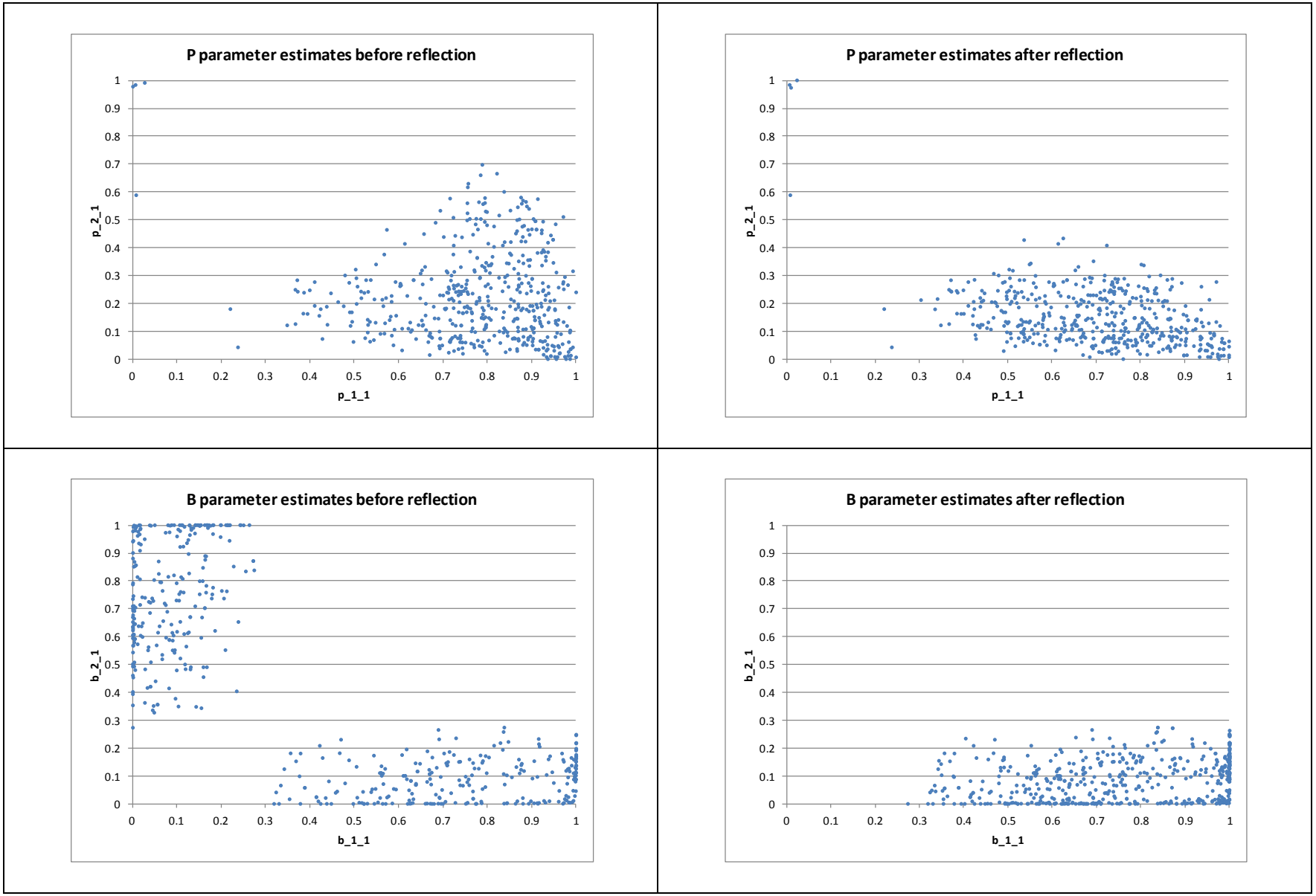


Figure 7.1.11: Scatter plot of the 500 BWA estimates contained in λ^* - using a signal sequence length of 250

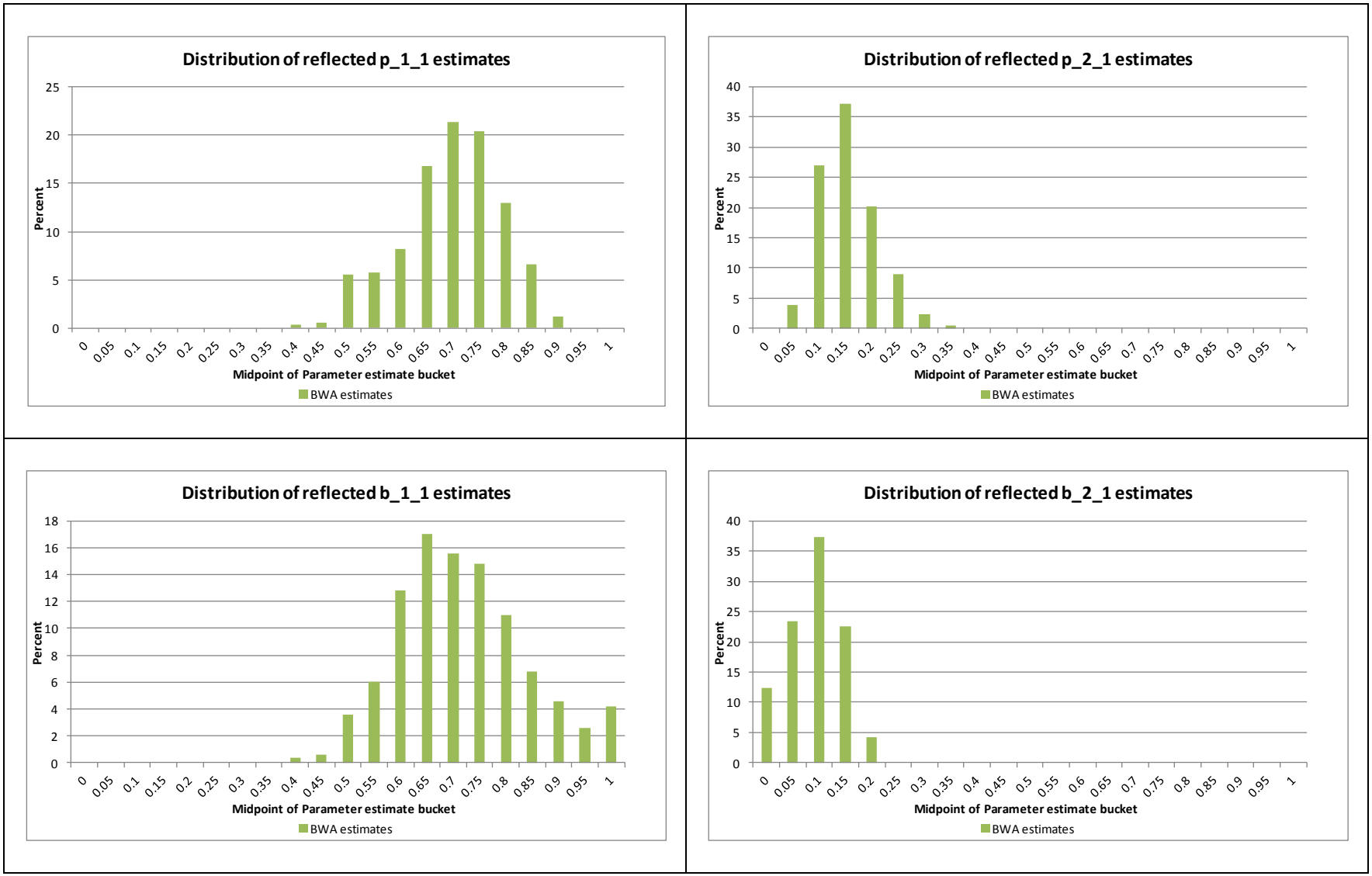


Figure 7.1.12: Frequency curves of the 500 BWA estimates contained in λ^* - using a signal sequence length of 1000

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

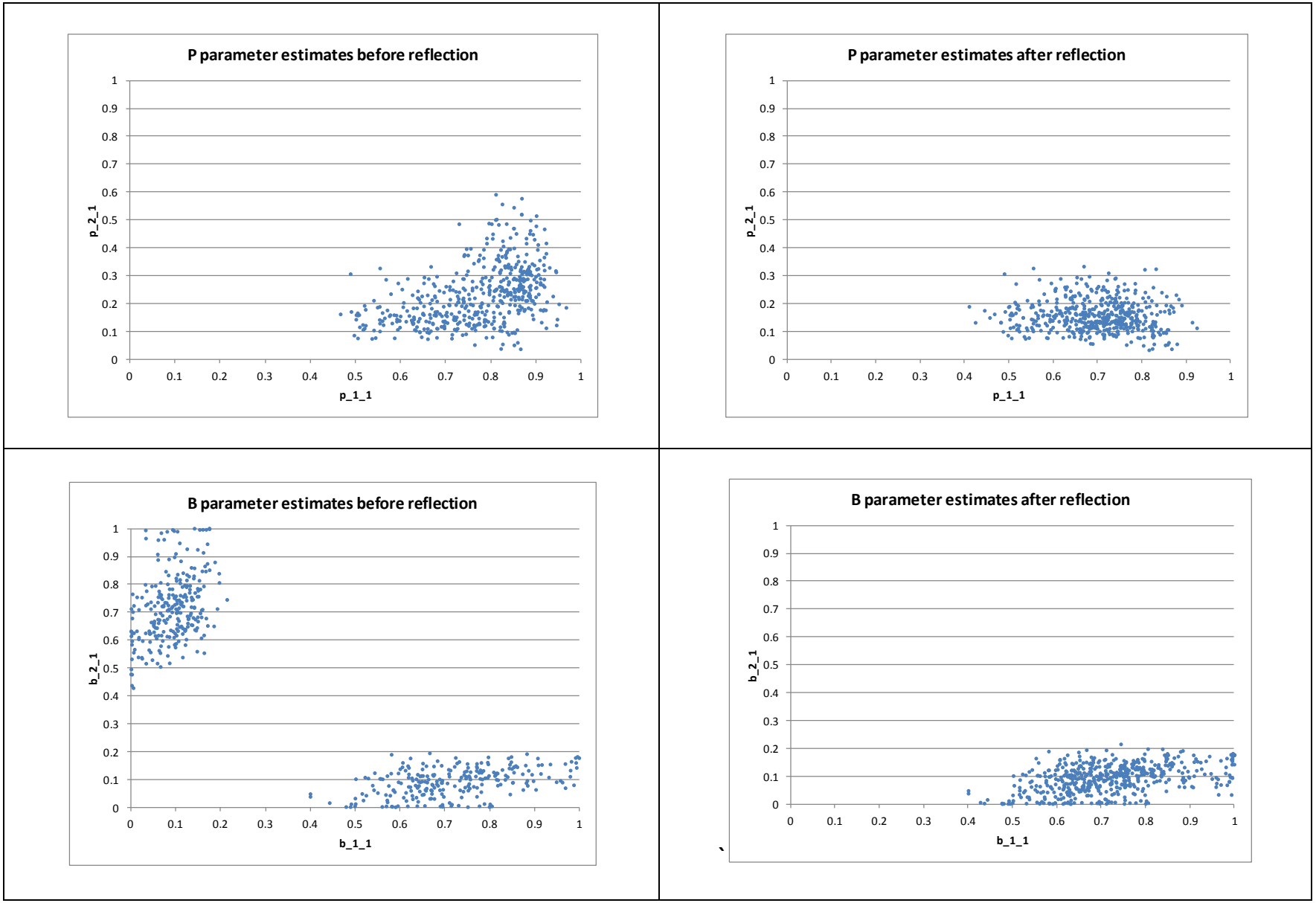


Figure 7.1.13: Scatter plot of the 500 BWA estimates contained in λ^* - using a signal sequence length of 1000

7.1.3 Concluding Remarks

The key findings from Sections 7.1.1 and 7.1.2 are summarised below. These findings were obtained by simulating data using the 5-parameter HMM specified by λ. It is however believed that the analysis described in this section can easily be used for HMMs with differing parameter values.

- The technique used to derive the BWA estimates in Table 7.1.1 may indeed prove useful in practice in estimating the true HMM parameters as the influence of the starting values used to train the BWA are taken into account. This is important as analysis confirmed that BWA estimates are indeed influenced by the starting values used to initialise the algorithm. This is somewhat expected as the BWA will typically locate a local maxima rather than a global maxima. The importance of considering appropriate reflection of the BWA estimates was also highlighted.

- Simulating multiple signal sequences allowed the statistical properties of the estimates in Table 7.1.1 to be investigated. In particular sampling distributions were compared when a signal sequence length of 250 and a signal sequence length of 1000 was used. The following was concluded from this analysis.

  - As expected, the final estimates calculated using the technique from Table 7.1.1 are indeed influenced by the data (signal sequence) used to train the algorithm. This gives rise to the sampling variability which can be observed in Figures 7.1.10, 7.1.11, 7.1.12 and 7.1.13 and Tables 7.1.2 and 7.1.3.

  - Using 150 random starting values for the BWA (generated using the uniform distribution) appears to be sufficient to ensure that the final estimates calculated using the technique from Table 7.1.1 is not significantly influenced by the starting values used for the BWA.

- The estimates for the P parameters (the transition probabilities for the underlying state sequence) showed little evidence of bias. Several of the simulated signal sequences led to the technique from Table 7.1.1 estimating B parameters (the probabilities of observing the signals given the states) at the boundary value, that is either zero or one. This resulted in the B parameters showing a degree of bias. It was however noted that using a signal sequence length of 1000 significantly diminished the number of B parameter estimates being estimated at a boundary value. This in turn lowered the bias of the B parameter estimates.
- The estimates calculated using the technique from Table 7.1.1 posses non-material variance (see Figures 7.1.10, 7.1.11, 7.1.12 and 7.1.13 and Tables 7.1.2 and 7.1.3). As mentioned above, this variability in the estimates is due to the dependence of the BWA on the data used to train the algorithm. It should however be noted that using a signal sequence length of 1000 significantly decreased the variance observed in the parameter estimates (by a factor of 3.4, 4.0, 2.6 and 2.3 for $p_{11}, p_{21}, b_{11}, b_{21}$ respectively).
- It is possible that the technique from Table 7.1.1 may result in an extreme outlying estimate. In particular 4 out of the 500 simulated runs (this equates to 0.8%) resulted in extreme outlying estimates for the P parameters when a signal sequence length of 250 was used to perform the simulations. No obvious outliers were however identified in the sampling distributions when a signal sequence length of 1000 was used to perform the simulations.

- The above analysis confirmed the expected result that using longer observed signal sequences to train the BWA yields more accurate final BWA estimates.

7.2 Exploring the Viterbi Algorithm for the HMM

7.2.1 Simulation Results

The use of the Viterbi Algorithm (VA) to predict the underlying hidden state sequence of a HMM was detailed in Section 3.2. This section will use the different simulation scenarios described in Section 7.1 to explore how effectively the VA recovers the underlying hidden state sequences which were simulated using the HMM $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$ defined in Section 7.1. It should be noted that scaling was needed in order to perform the VA due to the length of the signal sequences. For this purpose scaling using the natural logarithm, as described in [37], was used.

To begin, recall the parameter estimates for λ calculated using the BWA given in Table 7.1.1. To derive these estimates the BWA was trained separately using a simulated signal sequence of length of 75, 150, 250, 500, 1000 and 5500 respectively. Let these simulated signal sequences be denoted by $\{\mathbf{s}_{75}, \mathbf{s}_{150}, \mathbf{s}_{250}, \mathbf{s}_{500}, \mathbf{s}_{1000}, \mathbf{s}_{5500}\}$ and the corresponding BWA estimates of Table 7.1.1 be denoted by $\{\hat{\lambda}_{75}, \hat{\lambda}_{150}, \hat{\lambda}_{250}, \hat{\lambda}_{500}, \hat{\lambda}_{1000}, \hat{\lambda}_{5500}\}$. Due to the fact that each signal sequence $\{\mathbf{s}_{75}, \mathbf{s}_{150}, \mathbf{s}_{250}, \mathbf{s}_{500}, \mathbf{s}_{1000}, \mathbf{s}_{5500}\}$ was simulated, the corresponding underlying state sequence is in this case known, denoted by $\{\mathbf{x}_{75}, \mathbf{x}_{150}, \mathbf{x}_{250}, \mathbf{x}_{500}, \mathbf{x}_{1000}, \mathbf{x}_{5500}\}$. Hence, for a given signal sequence length $n \in \{75, 150, 250, 500, 1000, 5500\}$, $\mathbf{s}_n$ and $\hat{\lambda}_n$ can be used to perform the VA and estimate the state sequence, denoted by $\hat{\mathbf{x}}_n$. A comparison of $\hat{\mathbf{x}}_n$ and $\mathbf{x}_n$ then provides a measure of the accuracy of the Viterbi path. Of course λ (as opposed to $\hat{\lambda}_n$) can also be used to perform the VA and estimate the state sequence. This gives a measure as to whether the performance of the VA diminishes when estimated model parameters are used to perform the VA as opposed to the actual model parameters.

The exercise described above was performed and the results are given in Table 7.2.1. The results given are the simulated joint distribution of the true state i (rows) and the Viterbi estimate j (columns) of the state. The left tables represent the results

when λ was used to perform the VA, while the right tables represent the results when the BWA estimates $\{\hat{\lambda}_{75}, \hat{\lambda}_{150}, \hat{\lambda}_{250}, \hat{\lambda}_{500}, \hat{\lambda}_{1000}, \hat{\lambda}_{5500}\}$ from Table 7.1.1 were used to perform the VA. The percentage of states which were correctly predicted by the VA (i.e those which appear on the diagonals of the 2×2 tables) is also given.

The first result to note is that when λ was used to perform the VA, the Viterbi state path correctly predicted the true underlying state path for 80% to 90% of the n time points. This is true for all the signal sequence lengths $n \in \{75, 150, 250, 500, 1000, 5500\}$ which were tested. It is interesting to note that despite the BWA estimates being quite different to the true parameter values for some sequence lengths (see Table 7.1.1), when $\{\hat{\lambda}_{75}, \hat{\lambda}_{150}, \hat{\lambda}_{250}, \hat{\lambda}_{500}, \hat{\lambda}_{1000}, \hat{\lambda}_{5500}\}$ was used to perform the VA, the percentage of states correctly predicted was comparable to when λ was used to perform the VA, the biggest difference occurring for $n = 250$. Even in this instance however the VA correctly predicted the true state for 82% of the 250 time points (i.e. for 205 time points).

From these tables one may conclude that for $n = 5500$, for instance, the estimated probability that the inferred state is 2 if the true state is 2, is $\frac{3383}{5500}/\frac{3693}{5500}$ $= 0.916$ (when λ is used to perform the VA). More generally, Table 7.2.2 gives $P(\text{inferred state} = j|\text{true state} = i)$ and $P(\text{true state} = i|\text{inferred state} = j)$. This is done for each sequence length $n \in \{75, 150, 250, 500, 1000, 5500\}$ and is shown for when both λ and $\hat{\lambda}_n$ was used to perform the VA. Diagonal elements close to 1 are desirable for these tables. Here interestingly, $P(\text{inferred state} = 1|\text{true state} = 1)$ is consistently lower than $P(\text{inferred state} = 2|\text{true state} = 2)$; a possible reason for this could be that less data is available from state 1 (recall that the limiting steady state probabilities for this particular HMM are $\pi = [\frac{1}{3}, \frac{2}{3}]$). However, $P(\text{true state} = 1|\text{inferred state} = 1)$ shows higher probability values in most cases. This is encouraging since in practice the true state will typically not be known and will be estimated using the VA inferred state.

To further expand the above analysis, the simulated state and signal sequences from

Section 7.1.2 were also used to assess the effectiveness of the VA. Recall that in Section 7.1.2 a sequence length of $n = 250$ was used and 500 separate state sequences were simulated using the same HMM λ from Section 7.1.1. For each simulated state sequence, a signal sequence was simulated. The procedure described in Section 7.1.1 was performed for each simulated signal sequence to obtain a BWA estimate of λ. Hence 500 sets of BWA estimates were obtained (one set for each simulated signal sequence).

For each of the 500 simulated signal sequences, the VA was performed separately using both λ and the BWA estimate of λ associated with the signal sequence. This was then compared to the simulated state sequence to measure the percentage of states in the state sequence which were correctly predicted by the Viterbi path. These 500 percentages are shown in Figure 7.2.1, separately for when λ and the BWA estimate of λ was used to perform the VA. Let these distributions be denoted by P_1 and $\hat{P}_1$ respectively. In addition to this, there are two added plots in Figure 7.2.1. These were obtained by repeating the exercise described, but using a new set of 500 simulated state and signal sequences (while still using λ and the original BWA estimates of λ from Section 7.1.2 to perform the VA). Let these distributions of the percentage of states correctly predicted be denoted by P_2 and $\hat{P}_2$ respectively. [2]

It can be seen from Figure 7.2.1 that, as expected, the distribution of P_1 and P_2 are very similar with the majority of the percentage of states correctly predicted lying between 77.5% and 92.5%. The distribution of $\hat{P}_1$ lies slightly more to the right than the distribution of $\hat{P}_2$. This suggests that (for this particular HMM λ) the Viterbi path predicted the underlying state sequences slightly better when the same signal sequence which was used to train the BWA was also used to perform the VA. For both $\hat{P}_1$ and $\hat{P}_2$, the majority of the percentage of states correctly predicted was between

[2] In obtaining $\hat{P}_1$, the same simulated signal sequence which was used to train the BWA was also used (together with this BWA estimate) to perform the VA. This could lead to a potential bias in measuring the accuracy of the VA. This follows as using a BWA estimate in conjunction with the signal sequence used to train the BWA may produce a more accurate Viterbi path than if another signal sequence (not used to train the BWA) was used to perform the VA.

67.5% and 92.5%. These distributions lie to the left of P_1 and P_2 indicating that the VA better predicted the underlying state sequences when actual model parameters were used to perform the VA rather than the BWA estimates (a somewhat expected result). Also noticeable is the long lower tail of $\hat{P}_1$ and $\hat{P}_2$; that is for some of the simulation runs, the Viterbi path (when the BWA parameter estimates were used to train the VA) poorly predicted the true state path. For $\hat{P}_1$, the prediction accuracy of the Viterbi path was less than 67.5% for 46 of the 500 (9.2%) simulations; while for $\hat{P}_2$, the prediction accuracy of the Viterbi path was less than 67.5% for 63 of the 500 (12.6%) simulations. Analysis showed that all of the 46 simulation runs which were below 67.5% for $\hat{P}_1$ were also below 67.5% for $\hat{P}_2$. To further investigate this, the BWA estimates for these simulations (where the VA performed poorly) were isolated. Figure 7.2.2 shows the 46 sets of BWA estimates mentioned above (where $\hat{P}_1 < 67.5\%$). Two things are noticeable from Figure 7.2.2. Firstly the four sets of outlying BWA estimates which were identified in Section 7.1.2 are contained in the 46 isolated sets of BWA estimates. Secondly the remaining 42 sets of BWA estimates are clustered around the point $(\hat{p}_{11}, \hat{p}_{21}) = (0.95, 0.10)$ and $(\hat{b}_{11}, \hat{b}_{21}) = (0.40, 0.07)$. That is, for this particular HMM λ, when BWA estimates in the vicinity of $(p_{11}, p_{21}) = (0.95, 0.10)$ and $(b_{11}, b_{21}) = (0.40, 0.07)$ are used to train the VA, the success rate of the Viterbi path predicting the true state path is poor.

Next Figure 7.2.3 is presented which depicts the distributions of P_1, $\hat{P}_1$, P_2 and $\hat{P}_2$, but with the results from the 46 simulations mentioned above removed from P_1 and $\hat{P}_1$, and the results from the 63 simulations mentioned above removed from P_2 and $\hat{P}_2$. It can be seen from $\hat{P}_1$ and $\hat{P}_2$ in this graph that once the simulation runs mentioned above are removed, the Viterbi path (trained using BWA estimates) predicts the true state path well, although not quite as well as the Viterbi path trained using the actual model parameters. Once again the distribution of $\hat{P}_1$ lies slightly more to the right than the distribution of $\hat{P}_2$, suggesting that (for this particular HMM λ) the Viterbi path predicted the underlying state sequences slightly better when the same signal

sequence which was used to train the BWA was also used to perform the VA.

The importance of considering appropriate reflection of the BWA estimates was discussed in Section 7.1. The importance of this reflection can again be highlighted when BWA estimates are used to perform the VA. Figure 7.2.4 shows the distribution of $\hat{P}_1$ when the BWA parameter estimates used to perform the VA were not reflected where necessary. A comparison to the distribution of $\hat{P}_1$ depicted in Figure 7.2.1 (whereby reflection to the BWA parameters was applied before performing the VA) clearly indicates that the Viterbi path predicts the true state path significantly better if reflection is appropriately applied to the BWA estimated parameters before performing the VA.

Next the 500 simulated state and signal sequences of length 1000 from Section 7.1.2 were used to re-perform the above exercise. The corresponding P_1, $\hat{P}_1$, P_2 and $\hat{P}_2$ distributions are plotted in Figure 7.2.5. The paragraphs below highlight conclusions which can be drawn from these distributions.

Firstly, as was the case when a sequence length of 250 was used, the distributions of P_1 and P_2 are very similar with the majority of the percentage of states correctly predicted lying between 77.5% and 92.5%.

When analysing the prediction accuracy of $\hat{P}_1$ and $\hat{P}_2$ it is observed that the lower tail of these distributions is not as long as when the sequence length of 250 was used. For $\hat{P}_1$, only 5 of the 500 (1.0%) simulations had a Viterbi path prediction accuracy of less than 67.5%; while for $\hat{P}_2$, only 6 of the 500 (1.2%) simulations had a Viterbi path prediction accuracy of less than 67.5%. These are significantly less than the comparative results which were given earlier when a sequence length of 250 was used (9.2% and 12.6%). The second point to note is that for both $\hat{P}_1$ and $\hat{P}_2$ the majority of the percentage of states correctly predicted lies above 72.5% (in particular 97.6% for $\hat{P}_1$ and 97.2% for $\hat{P}_2$). This is significantly higher than the comparative results when a sequence length of 250 was used (84.6% for $\hat{P}_1$ and 80.4% for $\hat{P}_2$). The sig-

nificant improvement in the BWA estimates which were obtained using the sequence length of 1000 (when compared to the BWA estimates which were obtained when the sequence length of 250 was used) was noted in the previous section (see for example Figures 7.1.11 and 7.1.13). From the above two points it is clear that when these improved BWA estimates are used to perform the VA, the prediction accuracy of the Viterbi path increases. It should however be noted that while the prediction accuracy of the Viterbi path has significantly improved when using a sequence length of 1000, the distributions of $\hat{P}_1$ and $\hat{P}_2$ still lie to the left of P_1 and P_2. That is, when BWA estimates are used to perform the VA, the prediction accuracy of the Viterbi path is still less than when the actual model parameters are used to perform the VA.

The final point to note is that when a sequence length of 250 was used, the distribution of $\hat{P}_2$ was to the left of $\hat{P}_1$. This however is significantly less notable when a sequence length of 1000 is used.

7.2.2 Concluding Remarks

In conclusion of Section 7.2, the results from various simulation exercises analysing the prediction accuracy of the VA have been presented and discussed. These simulations have been performed using the HMM specified by λ in Section 7.1. As these simulation exercises help quantify the expected accuracy of the Viterbi path for a given HMM, they are recommended in applications of other HMMs where the interpretation of the Viterbi path is an important objective of the analysis.

	VA performed using actual model parameters				VA performed using BWA parameter estimates				
		***j* =1**	***j* =2**		**% Correct**		***j* =1**	***j* =2**		**% Correct**
n =75	***i* =1**	17	5	22		***i* =1**	16	6	22	
	***i* =2**	10	43	53		***i* =2**	8	45	53	
		27	48	75	**80.0%**		24	51	75	**81.3%**
n =150		***j* =1**	***j* =2**		**% Correct**		***j* =1**	***j* =2**		**% Correct**
	***i* =1**	36	21	57		***i* =1**	40	17	57	
	***i* =2**	7	86	93		***i* =2**	8	85	93	
		43	107	150	**81.3%**		48	102	150	**83.3%**
n =250		***j* =1**	***j* =2**		**% Correct**		***j* =1**	***j* =2**		**% Correct**
	***i* =1**	40	26	66		***i* =1**	29	37	66	
	***i* =2**	6	178	184		***i* =2**	8	176	184	
		46	204	250	**87.2%**		37	213	250	**82.0%**
n =500		***j* =1**	***j* =2**		**% Correct**		***j* =1**	***j* =2**		**% Correct**
	***i* =1**	110	44	154		***i* =1**	125	29	154	
	***i* =2**	30	316	346		***i* =2**	44	302	346	
		140	360	500	**85.2%**		169	331	500	**85.4%**
n =1000		***j* =1**	***j* =2**		**% Correct**		***j* =1**	***j* =2**		**% Correct**
	***i* =1**	186	135	321		***i* =1**	246	75	321	
	***i* =2**	54	625	679		***i* =2**	135	544	679	
		240	760	1000	**81.1%**		381	619	1000	**79.0%**
n =5500		***j* =1**	***j* =2**		**% Correct**		***j* =1**	***j* =2**		**% Correct**
	***i* =1**	1180	627	1807		***i* =1**	1206	601	1807	
	***i* =2**	310	3383	3693		***i* =2**	348	3345	3693	
		1490	4010	5500	**83.0%**		1554	3946	5500	**82.7%**

Table 7.2.1: Prediction results of the Viterbi path, using both actual model parameters and BWA estimated model parameters to perform the Viterbi algorithm.

	P(inferred state = *j* | true state = *i*)					
	VA performed using actual model parameters			VA performed using BWA parameter estimates		
		***j* = 1**	***j* = 2**		***j* = 1**	***j* = 2**
n = 75	***i* = 1**	77%	23%	***i* = 1**	73%	27%
	***i* = 2**	19%	81%	***i* = 2**	15%	85%
n = 150	***i* = 1**	63%	37%	***i* = 1**	70%	30%
	***i* = 2**	8%	92%	***i* = 2**	9%	91%
n = 250	***i* = 1**	61%	39%	***i* = 1**	44%	56%
	***i* = 2**	3%	97%	***i* = 2**	4%	96%
n = 500	***i* = 1**	71%	29%	***i* = 1**	81%	19%
	***i* = 2**	9%	91%	***i* = 2**	13%	87%
n = 1000	***i* = 1**	58%	42%	***i* = 1**	77%	23%
	***i* = 2**	8%	92%	***i* = 2**	20%	80%
n = 5500	***i* = 1**	65%	35%	***i* = 1**	67%	33%
	***i* = 2**	8%	92%	***i* = 2**	9%	91%

	P(true state = *i* | inferred state = *j*)					
	VA performed using actual model parameters			VA performed using BWA parameter estimates		
		***j* = 1**	***j* = 2**		***j* = 1**	***j* = 2**
	***i* = 1**	63%	10%	***i* = 1**	67%	12%
	***i* = 2**	37%	90%	***i* = 2**	33%	88%
	***i* = 1**	84%	20%	***i* = 1**	83%	17%
	***i* = 2**	16%	80%	***i* = 2**	17%	83%
	***i* = 1**	87%	13%	***i* = 1**	78%	17%
	***i* = 2**	13%	87%	***i* = 2**	22%	83%
	***i* = 1**	79%	12%	***i* = 1**	74%	9%
	***i* = 2**	21%	88%	***i* = 2**	26%	91%
	***i* = 1**	78%	18%	***i* = 1**	65%	12%
	***i* = 2**	23%	82%	***i* = 2**	35%	88%
	***i* = 1**	79%	16%	***i* = 1**	78%	15%
	***i* = 2**	21%	84%	***i* = 2**	22%	85%

Table 7.2.2: Conditional probabilities obtained when comparing the Viterbi path and the true state path (using both actual model parameters and BWA estimated model parameters to perform the Viterbi algorithm).

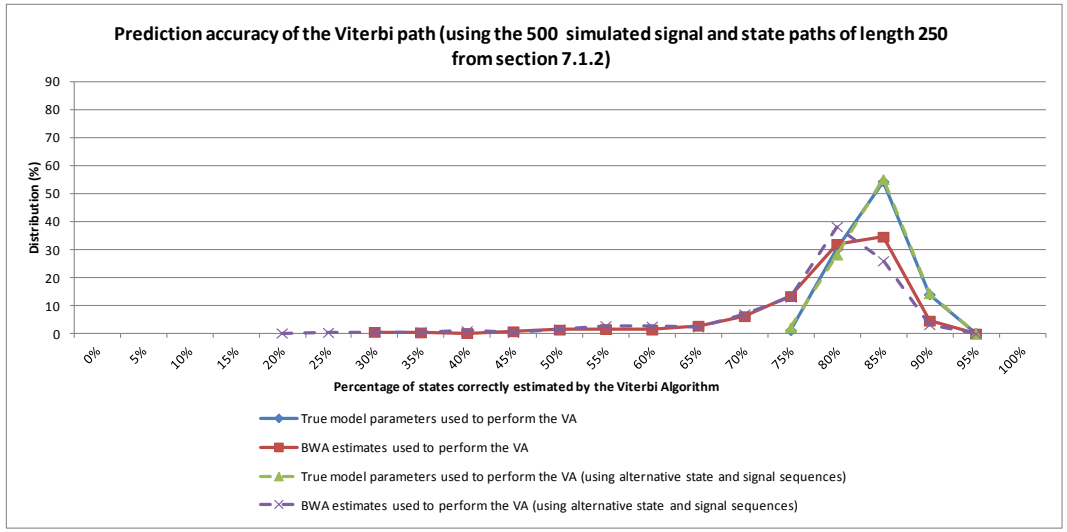


Figure 7.2.1: Prediction accuracy of the Viterbi path (using the 500 simulated signal and state paths of length 250 from section 7.1.2).*

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

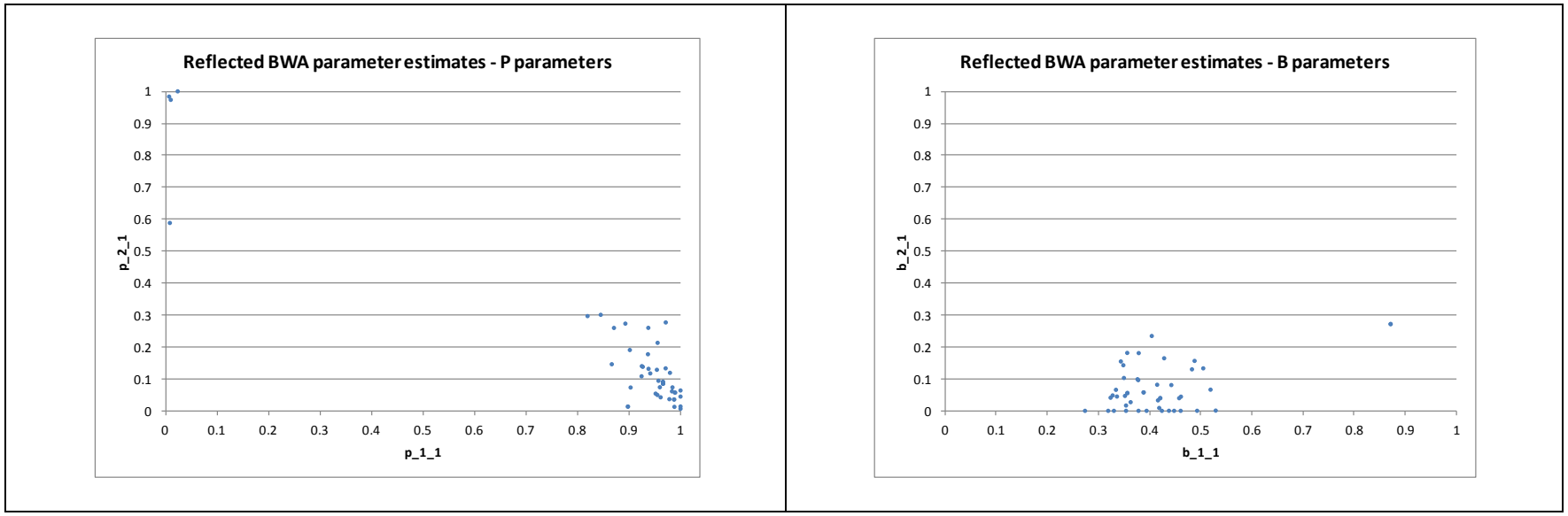


Figure 7.2.2: The BWA parameter estimates which, when used to perform the Viterbi Algorithm, led to poor prediction accuracy of the Viterbi path.

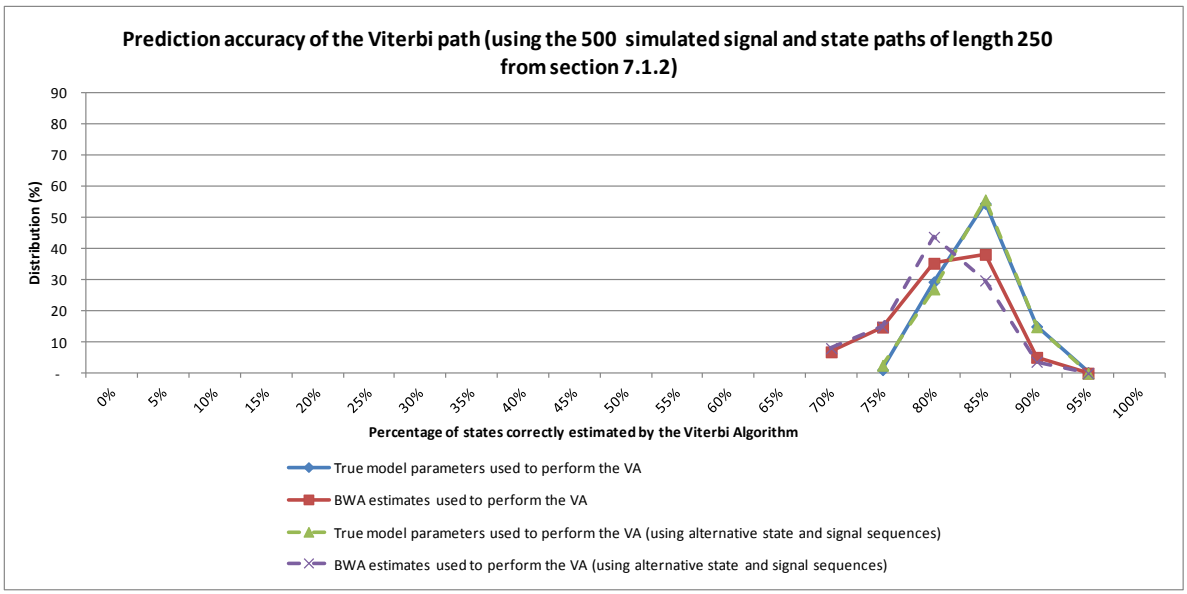


Figure 7.2.3: Prediction accuracy of the Viterbi path (using the simulated signal and state paths of length 250 from section 7.1.2). The simulations which led to the BWA estimates shown in figure 7.2.2 were removed. *

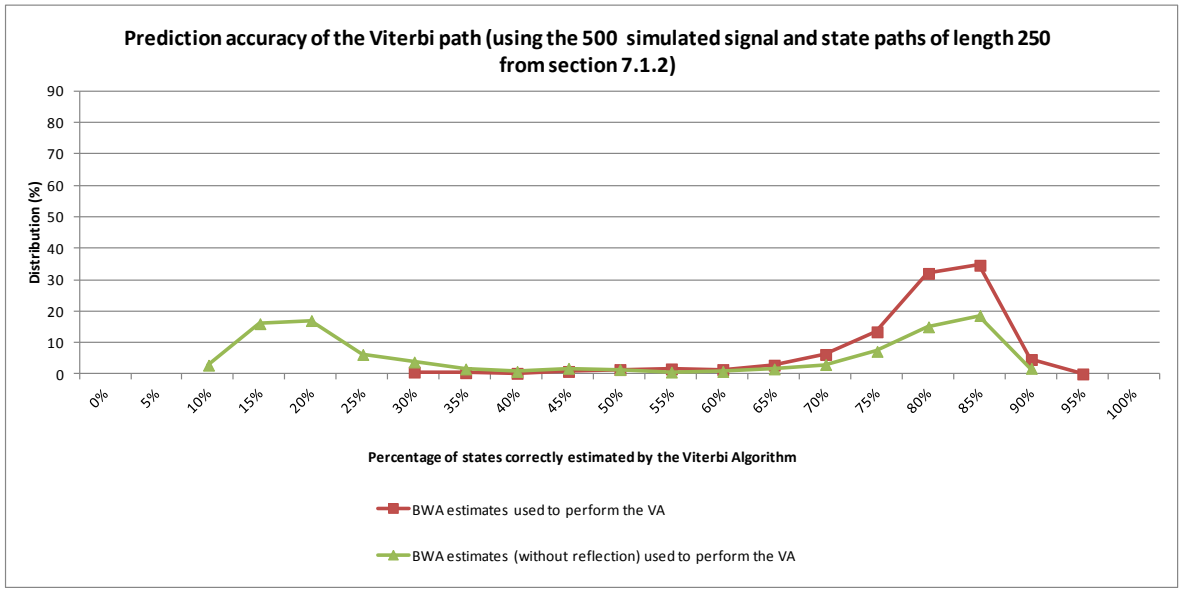


Figure 7.2.4: Prediction accuracy of the Viterbi path (using the 500 simulated signal and state paths of length 250 from section 7.1.2), including the scenario where un-reflected BWA parameter estimates were used to perform the VA. *

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

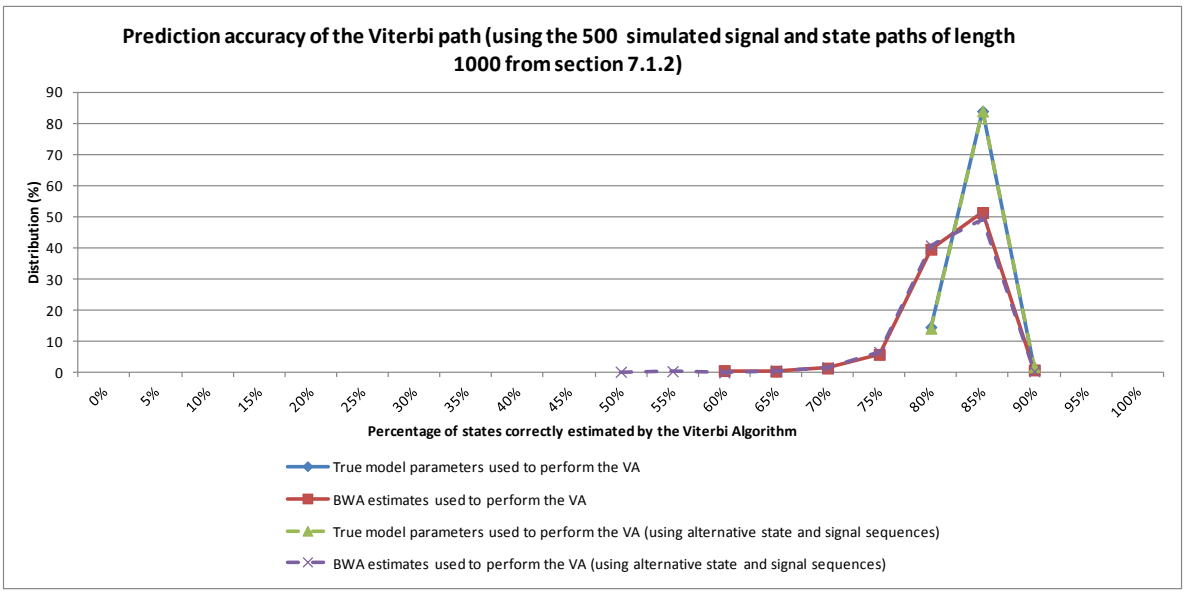


Figure 7.2.5: Prediction accuracy of the Viterbi path (using the 500 simulated signal and state paths of length 1000 from section 7.1.2). *

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

7.3 Exploring the Baum-Welch Algorithm for the DCMM

7.3.1 Simulation Results

Section 7.1 of this dissertation discussed a simulation exercise which highlighted properties of the BWA applied to a chosen HMM. In particular, in Section 7.1.1 it was shown for HMMs that the initial values used to train the BWA can greatly influence what the final BWA estimates will be. If this held true for the HMM, it is likely to also hold true for the DCMM. Analysis indeed confirms this. Hence the corresponding analysis from Section 7.1.1 will not be presented again for the DCMM. Instead this section will explore, through simulated sampling distributions, the properties of the BWA estimates for the DCMM (similar to what was discussed in Section 7.1.2 for the HMM), a study which is believed can provide a practitioner with valuable insights.

To begin, consider the following DCMM which will be used to perform the simulations for this section, $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$, where[3]

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 0.70 & 0.30 \\ 0.15 & 0.85 \end{pmatrix}$$

$$\mathbf{B}^{(1)} = \begin{pmatrix} b_{11}^{(1)} & b_{12}^{(1)} \\ b_{21}^{(1)} & b_{22}^{(1)} \end{pmatrix} = \begin{pmatrix} 0.60 & 0.40 \\ 0.05 & 0.95 \end{pmatrix}$$

$$\mathbf{B}^{(2)} = \begin{pmatrix} b_{11}^{(2)} & b_{12}^{(2)} \\ b_{21}^{(2)} & b_{22}^{(2)} \end{pmatrix} = \begin{pmatrix} 0.50 & 0.50 \\ 0.30 & 0.70 \end{pmatrix}. \tag{7.1}$$

This DCMM λ was chosen in order to ensure some degree of alignment to the HMM which was used to perform the analysis of Section 7.1. In particular, the same $\mathbf{a}$ and $\mathbf{P}$ parameters which were chosen for the HMM analysis are once again chosen for

[3]It should be noted that the results discussed in this section relate to the DCMM specified by λ. This DCMM is such that $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$ are similar to some degree, and is also such that most probabilities within $\mathbf{P}$, $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$ are not close to a boundary value. Sampling distributions of BWA parameter estimates may show more (or less) accuracy for other specified DCMMs. This may be explored for differing DCMMs through simulations similar to those presented in this section.

the DCMM analysis. Hence the limiting steady state probabilities for the underlying Markov chain is once again $\pi = [\frac{1}{3}\ ,\ \frac{2}{3}]$. Furthermore, the marginal probabilities of the HMM used in Section 7.1 outputting signal 1 and signal 2 were 0.3 and 0.7 respectively. The marginal probabilities of outputting signal 1 and signal 2 for λ specified in equation (7.1) are comparable. To show this the following analysis was performed. Using λ specified in equation (7.1), 100 separate state and signal sequences, each of length 1000, were simulated. The proportion of times signal 1 and signal 2 appeared was calculated for each of the 100 signal sequences. The mean and median of these 100 proportions was 0.307 and 0.309 respectively for signal 1; and 0.693 and 0.691 for signal 2. The corresponding mean and median proportions for the 100 state sequences was 0.332 and 0.334 for state 1 and 0.668 and 0.666 for state 2 (which are very comparable to the discussed theoretical limiting steady state probabilities $\pi = [\frac{1}{3}\ ,\ \frac{2}{3}]$).

In Section 7.1 the importance that reflection of the BWA estimates be considered for the HMM was discussed. Similarly reflection of the BWA estimates for the DCMM will also need consideration. To address this, consider the following three DCMMs.

$$\mathbf{a} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} 0.85 & 0.15 \\ 0.30 & 0.70 \end{pmatrix}$$
$$\mathbf{B}^{(1)} = \begin{pmatrix} 0.50 & 0.50 \\ 0.30 & 0.70 \end{pmatrix} \qquad \mathbf{B}^{(2)} = \begin{pmatrix} 0.60 & 0.40 \\ 0.05 & 0.95 \end{pmatrix} \tag{7.2}$$

$$\mathbf{a} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} 0.70 & 0.30 \\ 0.15 & 0.85 \end{pmatrix}$$
$$\mathbf{B}^{(1)} = \begin{pmatrix} 0.95 & 0.05 \\ 0.40 & 0.60 \end{pmatrix} \qquad \mathbf{B}^{(2)} = \begin{pmatrix} 0.70 & 0.30 \\ 0.50 & 0.50 \end{pmatrix} \tag{7.3}$$

$$\mathbf{a} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} 0.85 & 0.15 \\ 0.30 & 0.70 \end{pmatrix}$$
$$\mathbf{B}^{(1)} = \begin{pmatrix} 0.70 & 0.30 \\ 0.50 & 0.50 \end{pmatrix} \qquad \mathbf{B}^{(2)} = \begin{pmatrix} 0.95 & 0.05 \\ 0.40 & 0.60 \end{pmatrix}. \tag{7.4}$$

Examination of the DCMMs expressed in equations (7.2)-(7.4) reveal that they are equivalent in terms of statistical properties to the DCMM expressed in equation (7.1). All that has changed between the models is that the state and/or signal labels have been permuted. Analysis revealed that for different simulated signal sequences and initial parameter set inputs, the BWA interchangeably estimated the parameters associated with either of the DCMMs expressed in (7.1)-(7.4). Hence, in order to make meaningful interpretations from the simulation exercise which will be presented in this section, the parameters estimated by the BWA need to be appropriately reflected to represent the parameters of (7.1). This is achieved as follows.

Firstly, in order to make (7.2) identifiable to (7.1) and (7.4) identifiable to (7.3), all (p_{11}, p_{21}) estimates which lie above the diagonal $p_{21} = 1 - p_{11}$ need to be reflected around the diagonal $p_{21} = 1 - p_{11}$. This is achieved using the following logic:

$$
\begin{aligned}
&if \quad && p_{21} > 1 - p_{11} \\
&then \quad && p_{11} = 1 - p_{21} \\
& && p_{21} = 1 - p_{11} \\
& && b_{11}^{(1)} = b_{11}^{(2)} \\
& && b_{21}^{(1)} = b_{21}^{(2)} \\
& && b_{11}^{(2)} = b_{11}^{(1)} \\
& && b_{21}^{(2)} = b_{21}^{(1)} \,.
\end{aligned} \tag{7.5}
$$

As mentioned, performing the transformation given in equation (7.5) will ensure that the DCMMs in (7.1)-(7.4) are now expressed as either (7.1) or (7.3). In order to make (7.3) identifiable to (7.1), all $(b_{11}^{(1)}, b_{21}^{(1)})$ estimates which lie above the diagonal $b_{21}^{(1)} = 1 - b_{11}^{(1)}$ need to be reflected around the diagonal $b_{21}^{(1)} = 1 - b_{11}^{(1)}$; and similarly all $(b_{11}^{(2)}, b_{21}^{(2)})$ estimates which lie above the diagonal $b_{21}^{(2)} = 1 - b_{11}^{(2)}$ need to be reflected around the diagonal $b_{21}^{(2)} = 1 - b_{11}^{(2)}$. This is achieved using the following logic:

$$
\begin{aligned}
if \quad & b_{21}^{(1)} > 1 - b_{11}^{(1)} \\
then \quad & b_{11}^{(1)} = 1 - b_{21}^{(1)} \\
& b_{21}^{(1)} = 1 - b_{11}^{(1)} \\
\\
if \quad & b_{21}^{(2)} > 1 - b_{11}^{(2)} \\
then \quad & b_{11}^{(2)} = 1 - b_{21}^{(2)} \\
& b_{21}^{(2)} = 1 - b_{11}^{(2)} \, .
\end{aligned} \tag{7.6}
$$

Application of transformations (7.5) and (7.6) ensures that BWA parameter estimates associated with one of the DCMMs (7.2)-(7.4) are appropriately reflected to represent the parameters of the DCMM (7.1).

The approach followed to obtain simulated sampling distributions of the BWA estimates for the DCMM is similar to that which was used for the HMM in Section 7.1.2. This is outlined again for ease of reference.

To begin assume the DCMM $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$ which was specified in equation (7.1). Using λ, 500 distinct state and signal sequences were simulated, each of length 1000. Let these signal sequences be denoted by

$$\tilde{\mathbf{s}}_{1000}^{1}, \tilde{\mathbf{s}}_{1000}^{2}, \ldots, \tilde{\mathbf{s}}_{1000}^{500} \, .$$

For each simulated signal sequence $\tilde{\mathbf{s}}_{1000}^{k}$, where $k \in \{1, 2, \ldots 500\}$, 150 distinct starting values for the BWA were randomly generated from the uniform(0,1) distribution such that the required probability properties hold. Let these be denoted by

$$\tilde{\lambda}_k^{150} = \{\tilde{\lambda}_k^{(1)}, \tilde{\lambda}_k^{(2)} \ldots, \tilde{\lambda}_k^{(150)}\} \, .$$

In other words, for each $k = 1, 2, \ldots, 500$ a signal sequence of length of 1000, $\tilde{\mathbf{s}}_{1000}^{k}$, is simulated and the BWA is performed 150 times, using the 150 randomly generated

distinct starting values contained in $\tilde{\lambda}_k^{150}$. Thus for each k, 150 distinct BWA estimates are calculated, denoted by

$$\hat{\lambda}_k^{150} = \{\hat{\lambda}_k^{(1)}, \hat{\lambda}_k^{(2)} \ldots, \hat{\lambda}_k^{(150)}\}\,.$$

The likelihood value for each of the BWA parameter set estimates within $\hat{\lambda}_k^{150}$ is then calculated; that is for each $i = 1, 2, \ldots, 150$ the following is calculated:

$$l_k^{(i)} = P(\mathbf{S}_{1000} = \tilde{\mathbf{s}}_{1000}^k | \hat{\lambda}_k^{(i)})\,.$$

Let the BWA estimate corresponding to the maximum of $\{l_k^{(1)}, l_k^{(2)}, \ldots, l_k^{(150)}\}$ be denoted by λ_k^*. That is

$$\lambda_k^* = \arg\max_{\hat{\lambda}_k^{(i)}} \{l_k^{(i)} \,:\, i = 1, 2, \ldots, 150\}\,.$$

This is repeated for each k, yielding 500 final BWA estimates calculated using 500 distinct simulated signal sequences and 150 distinct initial inputs into the BWA for each of the 500 simulated signal sequences. Let this be denoted as

$$\lambda^* = \{\lambda_1^*, \lambda_2^*, \ldots, \lambda_{500}^*\}.$$

Thus, λ^* is a (simulated) set of realisations from the sampling distribution of $\hat{\lambda}$ when the sequence length is 1000. Assessment of the distribution of λ^* gives light into the sensitivity of the BWA to the signal sequence used to train it. Importantly assessment of the distribution of λ^* will also give insight into the statistical properties of the BWA parameter estimates.

What may also be of interest to a practitioner is how the distribution of λ^* might change if a longer signal sequence is used to train the BWA. To this end, the above exercise was repeated using a signal sequence of length 2500 and the resulting λ^* investigated. In particular, graphs at the end of this section show the following (note that all parameter estimates shown in these graphs have been appropriately reflected according to equations (7.5) and (7.6)):

- Histograms of the 500 estimated BWA parameter estimates for p_{11} and p_{21} (Figure 7.3.1) and $b_{11}^{(1)}$, $b_{21}^{(1)}$, $b_{11}^{(2)}$ and $b_{21}^{(2)}$ (Figure 7.3.2) when signal sequences of length 1000 were used to train the BWA.
- Scatter plots showing the interaction between the 500 estimated BWA parameter estimates for p_{11} and p_{21}, $b_{11}^{(1)}$ and $b_{21}^{(1)}$, and $b_{11}^{(2)}$ and $b_{21}^{(2)}$ (Figure 7.3.3) when signal sequences of length 1000 were used to train the BWA.
- Histograms of the 500 estimated BWA parameter estimates for p_{11} and p_{21} (Figure 7.3.4) and $b_{11}^{(1)}$, $b_{21}^{(1)}$, $b_{11}^{(2)}$ and $b_{21}^{(2)}$ (Figure 7.3.5) when signal sequences of length 2500 were used to train the BWA.
- Scatter plots showing the interaction between the 500 estimated BWA parameter estimates for p_{11} and p_{21}, $b_{11}^{(1)}$ and $b_{21}^{(1)}$, and $b_{11}^{(2)}$ and $b_{21}^{(2)}$ (Figure 7.3.6) when signal sequences of length 2500 were used to train the BWA.
- Figures 7.3.7 and 7.7.8 will be discussed in more detail in the paragraphs below.
- Histograms comparing the distributions of the 500 estimated BWA parameter estimates when signal sequences of length 1000 and 2500 were used to train the BWA - these are shown for p_{11} and p_{21} (Figure 7.3.9) and $b_{11}^{(1)}$, $b_{21}^{(1)}$, $b_{11}^{(2)}$ and $b_{21}^{(2)}$ (Figure 7.3.10).

The two tables below also show the MSE analysis for the 500 BWA parameter estimates (after they have been appropriately reflected according to equations (7.5) and (7.6)) when signal sequences of length 1000 and 2500 were used to train the BWA.

Parameter	MSE(λ^*)	mean(λ^*)	Bias(λ^*, λ)^2	Var(λ^*)
p_{11}	0.151	0.568	0.017	0.133
p_{21}	0.122	0.257	0.011	0.110
$b_{11}^{(1)}$	0.140	0.337	0.069	0.070
$b_{21}^{(1)}$	0.075	0.220	0.029	0.046
$b_{11}^{(2)}$	0.023	0.510	0.000	0.022
$b_{21}^{(2)}$	0.022	0.184	0.014	0.008

Table 7.3.1: Mean squared error analysis of λ^*, when signal sequences of length of

1000 were used to train the BWA

Parameter	MSE(λ^*)	mean(λ^*)	Bias(λ^*, λ)^2	Var(λ^*)
p_{11}	0.115	0.583	0.014	0.102
p_{21}	0.094	0.237	0.008	0.087
$b_{11}^{(1)}$	0.121	0.348	0.063	0.058
$b_{21}^{(1)}$	0.088	0.251	0.040	0.047
$b_{11}^{(2)}$	0.020	0.507	0.000	0.020
$b_{21}^{(2)}$	0.024	0.178	0.015	0.009

Table 7.3.2: Mean squared error analysis of λ^*, when signal sequences of length of 2500 were used to train the BWA

Beginning with the **P** parameter estimates it can be seen that the estimates lie in two distinct groupings, one where $\hat{p}_{11} > \hat{p}_{21}$ and the other grouping where $\hat{p}_{11} < \hat{p}_{21}$. This is true for when both sequences of length 1000 and 2500 were used to train the BWA. As the true model parameters are $(p_{11}, p_{21}) = (0.70, 0.15)$, parameter estimates in the grouping $\hat{p}_{11} > \hat{p}_{21}$ are desirable. As such, it is hoped that as the length of the signal sequence used to train the BWA increases, the proportion of BWA estimates which are estimated in the region $\hat{p}_{11} < \hat{p}_{21}$ decreases. This is indeed the case - when a signal sequences of length 1000 were used to train the BWA, the proportion of estimates in the region $\hat{p}_{11} < \hat{p}_{21}$ was 30.0% (150 of the 500 estimates). This proportion decreased to 25.4% (127 of the 500 estimates) when signal sequences of length 2500 were used to train the BWA. Despite the decrease, this proportion is still concerning.

The sampling distributions in Figures 7.3.9 and 7.3.10 as well as the MSE analysis in Tables 7.3.1 and 7.3.2 can be used to assess how accurately the BWA estimates predict the true model parameter. The bias and variance in the **P** parameter estimates is material. This is somewhat expected due to the two distinct groupings for the **P** parameter estimates mentioned above. However, as the length of the signal sequence used to train the BWA is increased, it is hoped that the bias and variance shown in the sampling distribution will decrease. This is indeed the case, using signal sequences of length 2500 to train the BWA decreased the variance for the p_{11} and p_{21} parame-

ters (when compared to using signal sequences of length 1000) by factors of 1.31 and 1.27 respectively. The squared bias decreased by factors of 1.27 and 1.49 respectively and the mean squared error decreased by factors of 1.31 and 1.29 respectively. The improvement in accuracy of the $\mathbf{P}$ parameter estimates can also be seen visually in Figure 7.3.9. Another encouraging observation is that the BWA seems less likely to produce estimates on the boundary $(p_{11}, p_{21}) = (0,1)$ and $(p_{11}, p_{21}) = (1,0)$ when a sequence length of 2500 is used (see Figures 7.3.3, 7.3.6 and 7.3.9). However, despite these observed improvements when using sequences of length 2500, the variance and bias (and therefore also the MSE) is still material. As mentioned this is largely due to the grouping of estimates in the region $\hat{p}_{11} < \hat{p}_{21}$.

Analysis of the BWA estimates for the $b_{11}^{(1)}$ and $b_{21}^{(1)}$ parameters reveals that the BWA has not performed well in estimating the true parameter value for these parameters. In particular, the spread of the estimates is large (with little clustering at the true parameter values) leading to bias and variance in the estimates. The accuracy of the BWA estimates for $b_{11}^{(1)}$ appears to marginally improve when the length of the signal sequences used to train the BWA is increased from 1000 to 2500 (MSE decreases by a factor of 1.15). Increasing the signal length does not appear to yield noticeable improvement in the BWA estimates for $b_{21}^{(1)}$. One encouraging observation however is that the BWA seems less likely to produce estimates on the boundary $(b_{11}^{(1)}, b_{21}^{(1)}) = (0,0)$ when a sequence length of 2500 is used (see Figures 7.3.3, 7.3.6 and 7.3.10).

Analysis of the BWA estimates for the $b_{11}^{(2)}$ and $b_{21}^{(2)}$ parameters reveals that for this particular DCMM, the BWA has produced estimates which show greater accuracy in predicting the true parameter value than the estimates for $b_{11}^{(1)}$ and $b_{21}^{(1)}$. This is clear from the tables and graphs supplied, see for example Figures 7.3.3, 7.3.6 and 7.3.10 and Tables 7.3.1 and 7.3.2. Notably, the estimates for $b_{11}^{(2)}$ show very little bias and the variance and mean squared error decreases by a factor 1.15 as the length of the signal sequences used to train the BWA changes from 1000 to 2500. The improvement in the estimates can also be observed in the sampling distribution in Figure

7.3.10. The estimates for $b_{21}^{(2)}$ show more bias but less variance when compared to the estimates for $b_{11}^{(2)}$. Interestingly, the sampling distribution for $b_{21}^{(2)}$ does not appear to fundamentally change when the signal sequence length is increased from 1000 to 2500. A pleasing attribute of both the $b_{11}^{(2)}$ and $b_{21}^{(2)}$ estimates is that there does not appear to be an obvious clustering of the estimates on any of the boundary values.

For this particular DCMM, the BWA estimates for the $b_{11}^{(2)}$ and $b_{21}^{(2)}$ parameters appear to be more accurate than the BWA estimates for the $b_{11}^{(1)}$ and $b_{21}^{(1)}$ parameters (as was mentioned above). A possible reason for this could be that the limiting steady state probabilities for the underlying Markov chain is $\pi = [\frac{1}{3}, \frac{2}{3}]$. On average, over the course of a given state sequence, the underlying state is expected to be state 2 twice as often as it is expected to be state 1. Hence it is expected that there will be twice as much data available to estimate $b_{11}^{(2)}$ and $b_{21}^{(2)}$ when compared to $b_{11}^{(1)}$ and $b_{21}^{(1)}$. This is similar to the sampling distributions which were seen for the HMM - see Section 7.1.2. In this section, the HMM used had the same limiting steady state probabilities for the underlying Markov chain, $\pi = [\frac{1}{3}, \frac{2}{3}]$. For this HMM, the sampling distributions for the BWA estimates revealed that the BWA produced more accurate estimates (less bias and variance in the estimates) for b_{21} (signal probability given state 2) than for b_{11} (signal probability given state 1). The above explanation could potentially also be the reason why the estimates for p_{21} show a lower MSE than the estimates for p_{11} for both the HMM and the DCMM.

It is also insightful to compare the sampling distributions for the BWA estimates for the HMM to those of the DCMM. In particular, as mentioned at the start of this section, the two state, two signal HMM used in Section 7.1.2 is comparable to the two state, two signal DCMM used in this section. The results in Table 7.1.3 summarise the sampling distributions for the HMM BWA estimates when signal sequences of length 1000 were used to train the BWA. This can be compared to Table 7.3.1 of this section which summarises the sampling distributions for the DCMM BWA estimates when signal sequences of length 1000 were used to train the BWA. This comparison

reveals that the BWA estimates for the HMM estimated the true model parameters with notably more accuracy (for example, the highest MSE in Table 7.1.3 is lower than the lowest MSE in Table 7.3.1). This difference in estimation accuracy is also confirmed by comparing the figures in Section 7.1.2 with the figures of this section. That is, the added dependence between the DCMM signal outputs is likely to result in less accurate BWA estimates when compared to the BWA estimates for the HMM (provided of course that the length of the signal sequence used to train the BWA is the same for both the HMM and the DCMM).

Finally, it was noted earlier in this section that the $\mathbf{P}$ parameter estimates for this particular DCMM simulation exercise are such that the estimates lie in two distinct groupings, one where $\hat{p}_{11} > \hat{p}_{21}$ and the other grouping where $\hat{p}_{11} < \hat{p}_{21}$. What may be of interest is to investigate how the BWA estimates for the $b_{11}^{(1)}$, $b_{21}^{(1)}$, $b_{11}^{(2)}$ and $b_{21}^{(2)}$ differ for when the BWA estimates for p_{11} and p_{21} were estimated such that $\hat{p}_{11} > \hat{p}_{21}$ compared to when they were estimated such that $\hat{p}_{11} < \hat{p}_{21}$. This is shown in Figure 7.3.7 (when $\hat{p}_{11} > \hat{p}_{21}$) and Figure 7.3.8 (when $\hat{p}_{11} < \hat{p}_{21}$) for the case when sequences of length 2500 were used to train the BWA. The two tables below show the MSE analysis for the BWA estimates when signal sequences of length 1000 (Table 7.3.3) and 2500 (Table 7.3.4) were used to train the BWA and when $\hat{p}_{11} > \hat{p}_{21}$. Table 7.3.4 is then the MSE analysis for the BWA parameter estimates depicted in Figure 7.3.7. As expected, the MSE in Tables 7.3.3 and 7.3.4 is considerably lower for the $\mathbf{P}$ parameter estimates when compared to Tables 7.3.1 and 7.3.2

Parameter	MSE(λ^*)	mean(λ^*)	Bias(λ^*, λ)^2	Var(λ^*)
p_{11}	0.033	0.789	0.008	0.025
p_{21}	0.013	0.060	0.008	0.005
$b_{11}^{(1)}$	0.159	0.305	0.087	0.072
$b_{21}^{(1)}$	0.084	0.228	0.032	0.052
$b_{11}^{(2)}$	0.020	0.485	0.000	0.020
$b_{21}^{(2)}$	0.018	0.197	0.011	0.007

Table 7.3.3: Mean squared error analysis of λ^*, when signal sequences of length of

1000 were used to train the BWA and where $\hat{p}_{11} > \hat{p}_{21}$

Parameter	MSE(λ^*)	mean(λ^*)	Bias(λ^*, λ)^2	Var(λ^*)
p_{11}	0.023	0.754	0.003	0.020
p_{21}	0.011	0.081	0.005	0.006
$b_{11}^{(1)}$	0.139	0.316	0.081	0.058
$b_{21}^{(1)}$	0.103	0.267	0.047	0.056
$b_{11}^{(2)}$	0.019	0.489	0.000	0.018
$b_{21}^{(2)}$	0.020	0.191	0.012	0.009

Table 7.3.4: Mean squared error analysis of λ^*, when signal sequences of length of 2500 were used to train the BWA and where $\hat{p}_{11} > \hat{p}_{21}$

7.3.2 Concluding Remarks

The results from a simulation exercise which explored the sampling distributions of parameter estimates for a two-state two-signal DCMM were discussed in this section. The model parameters were estimated using the BWA. This was done separately for when signal sequences of length 1000 and 2500 were used to the train the BWA. In particular the accuracy of the BWA parameter estimates for six parameters of interest was investigated, namely p_{11}, p_{21}, $b_{11}^{(1)}$, $b_{21}^{(1)}$, $b_{11}^{(2)}$ and $b_{21}^{(2)}$. Conclusions drawn from this study are summarised below. While these conclusions relate to the specified DCMM which was used for the simulation study, it is believed that the analysis described can easily be replicated for DCMMs with differing parameter values.

- Reflection of parameter estimates produced by the BWA needs to be considered before meaningful interpretation can be made. A procedure to appropriately reflect parameter estimates for the DCMM was proposed in equations (7.5) and (7.6). Where necessary, this approach was used to reflect the parameter estimates presented in this section.

- As expected, the accuracy of the BWA parameter estimates is not equivalent across all six parameters. In particular, the sampling distribution of the p_{21}

estimates showed greater accuracy (according to the mean squared error) than the sampling distribution of the p_{11} estimates; and the sampling distributions of the $b_{11}^{(2)}$ and $b_{21}^{(2)}$ estimates showed greater accuracy than the sampling distributions of the $b_{11}^{(1)}$ and $b_{21}^{(1)}$ estimates. A possible reason for this could be that the limiting steady state probabilities for the underlying Markov chain is $\pi = [\frac{1}{3}\ ,\ \frac{2}{3}]$. On average, over the course of a given state sequence, the underlying state is expected to be in state 2 twice as often as it is expected to be in state 1. Hence it is expected that there will be twice as much data available to estimate p_{21}, $b_{11}^{(2)}$ and $b_{21}^{(2)}$ when compared to p_{11}, $b_{11}^{(1)}$ and $b_{21}^{(1)}$.

- Increasing the length of the signal sequence used to train the BWA from 1000 to 2500 appeared to improve the accuracy of some of the parameter estimates, but not all. This improvement was evident for the p_{11}, p_{21}, $b_{11}^{(1)}$ and $b_{11}^{(2)}$ estimates. Improvement in the accuracy of the parameter estimates for $b_{21}^{(1)}$ and $b_{21}^{(2)}$ was however not as apparent.

- The BWA estimates of some parameters exhibited a high MSE, even if signal sequences of length 2500 were used to train the BWA. This was particularly true for p_{11}, p_{21}, $b_{11}^{(1)}$ and $b_{21}^{(1)}$. Investigation revealed that the p_{11} and p_{21} parameter estimates were such that two distinct groupings were apparent in the estimates, one where $\hat{p}_{11} > \hat{p}_{21}$ and the other grouping where $\hat{p}_{11} < \hat{p}_{21}$. As the true model parameters are $(p_{11}, p_{21}) = (0.70, 0.15)$, a grouping of parameter estimates such that $\hat{p}_{11} < \hat{p}_{21}$ is concerning. Furthermore, investigation revealed that the variance of the $b_{11}^{(1)}$ and $b_{21}^{(1)}$ parameter estimates was considerable. This resulted in the mode of their sampling distributions being less identifiable than what would be desired.

  Based on these findings, it is recommended that either a signal sequence longer than 2500 points or multiple signal sequences be used in practice to train the BWA and estimate the parameters for a two-state two signal DCMM.

- When signal sequences of length 1000 were used to train the BWA, the parameter estimates for the HMM (see Section 7.1.2) showed greater accuracy than the parameter estimates for the DCMM. This is somewhat expected due to the additional dependence structure in the signal outputs of the DCMM. This suggests that, in order to produce accurate parameter estimates, a longer signal sequence is required to train the BWA for the DCMM than for the HMM.

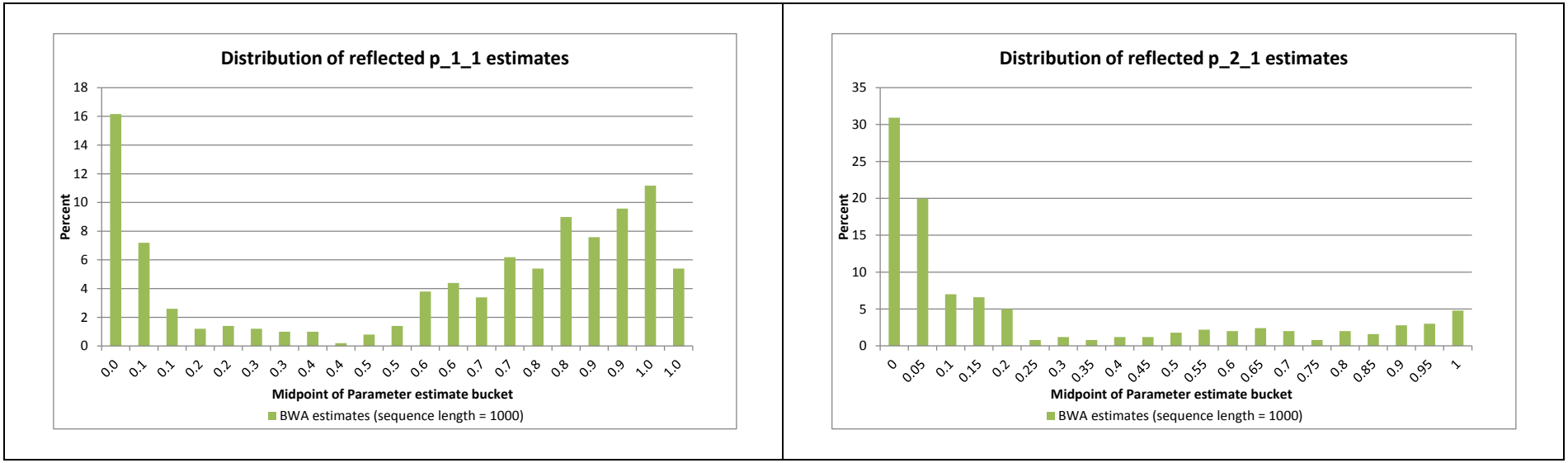


Figure 7.3.1: Frequency curves of the 500 BWA estimates (for the state transition probabilities) contained in λ^* - using a signal sequence length of 1000.

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025, 0.025) to [0,975, 1.025)

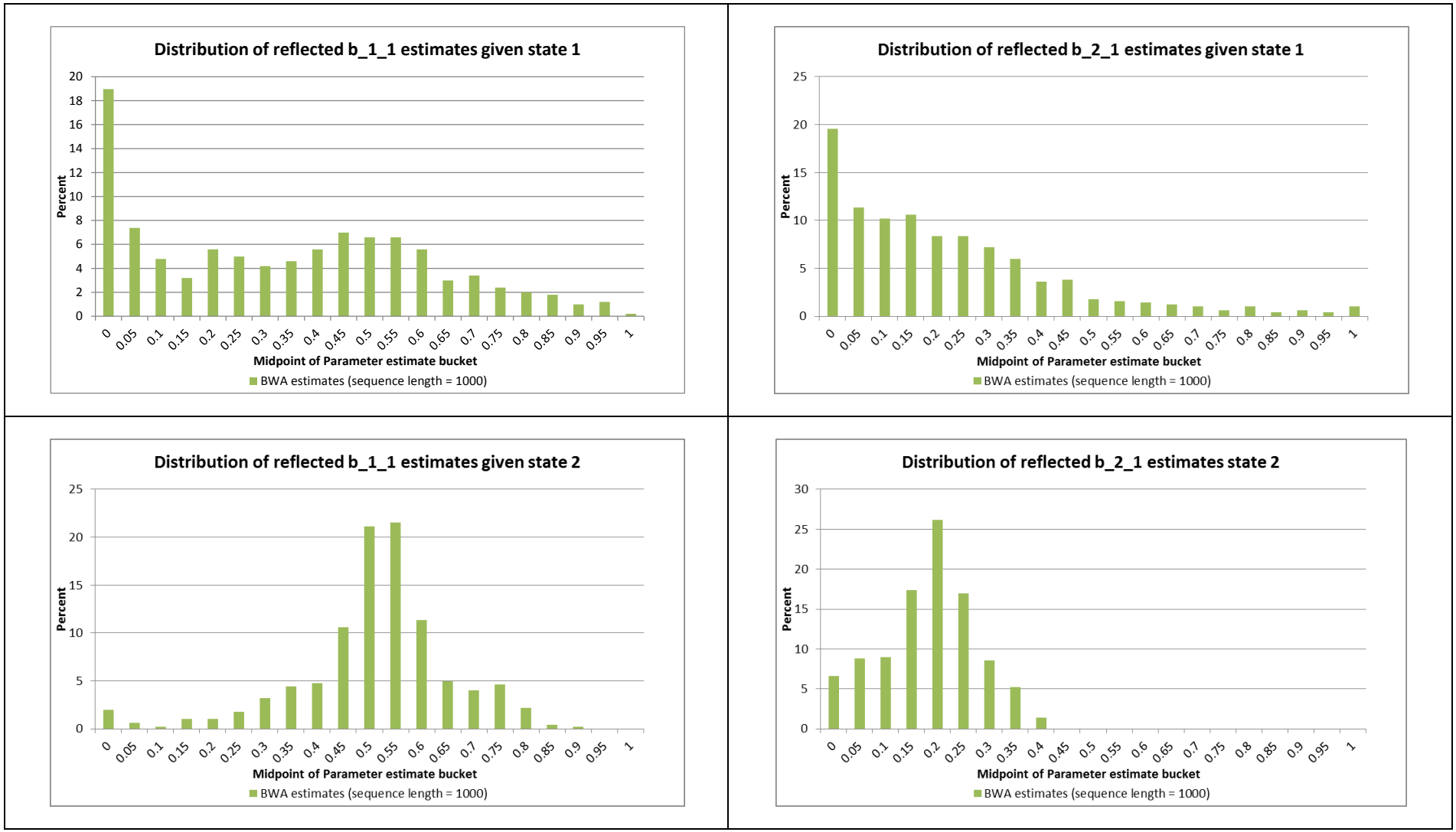
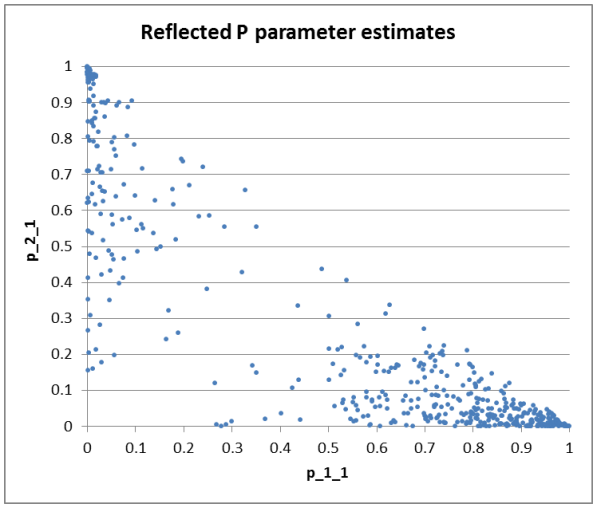
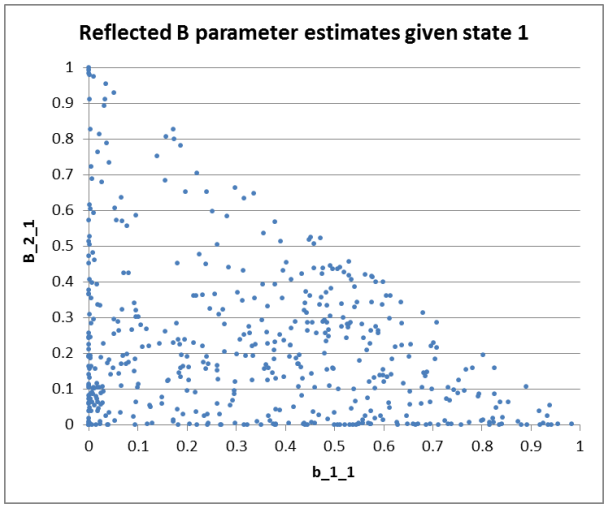


Figure 7.3.2: Frequency curves of the 500 BWA estimates (for the signal transition probabilities) contained in λ^* - using a signal sequence length of 1000.

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025, 0.025) to [0,975, 1.025)





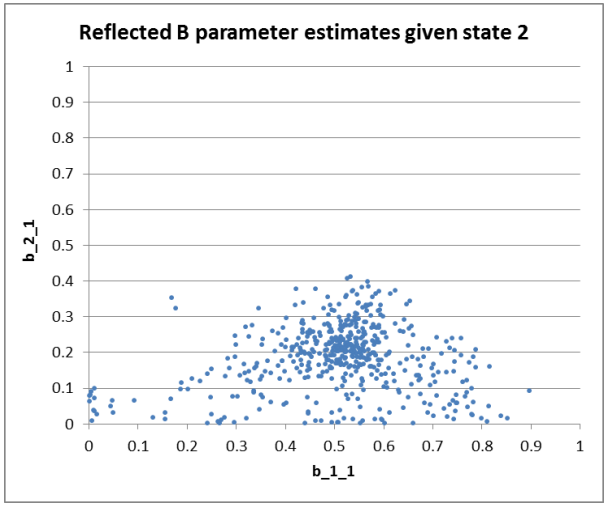


Figure 7.3.3: Scatter plot of the 500 BWA estimates contained in λ^* - using a signal sequence length of 1000.

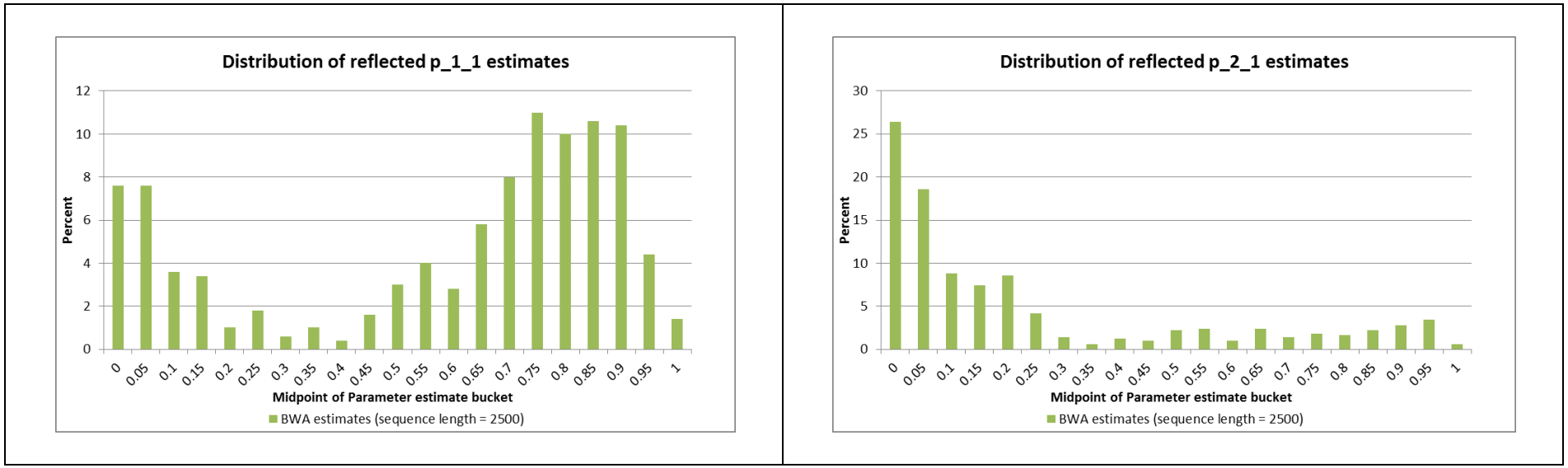


Figure 7.3.4: Frequency curves of the 500 BWA estimates (for the state transition probabilities) contained in λ^* - using a signal sequence length of 2500.

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025, 0.025) to [0,975, 1.025)

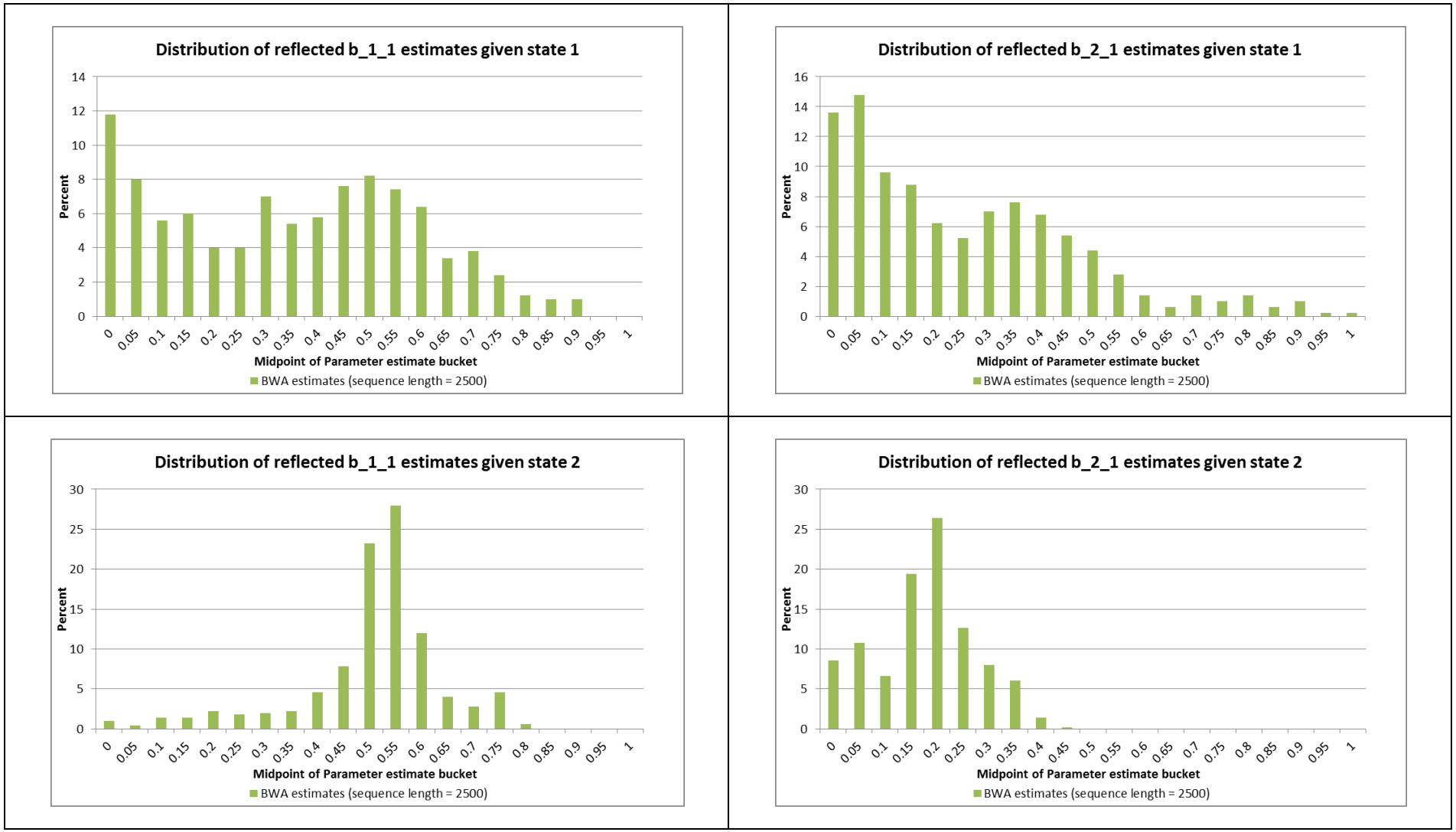
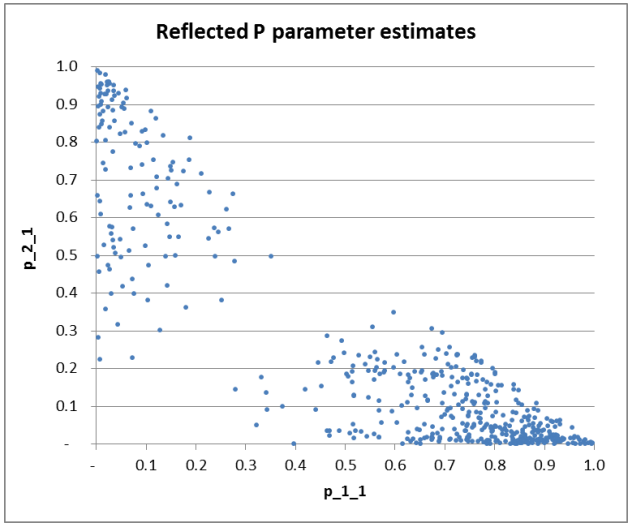
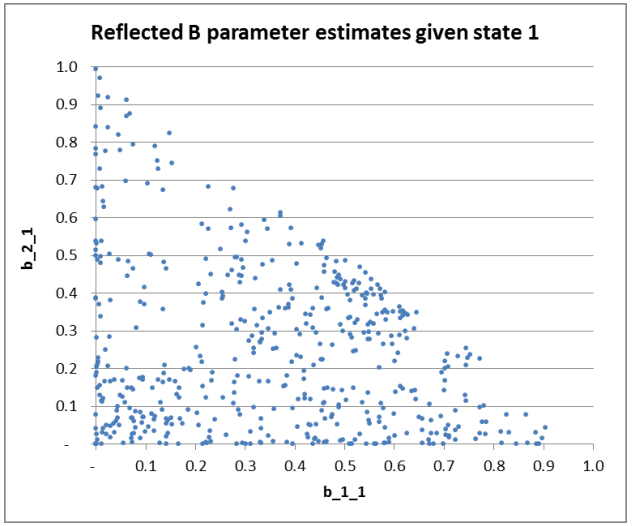


Figure 7.3.5: Frequency curves of the 500 BWA estimates (for the signal transition probabilities) contained in λ^* - using a signal sequence length of 2500.

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025, 0.025) to [0,975, 1.025)





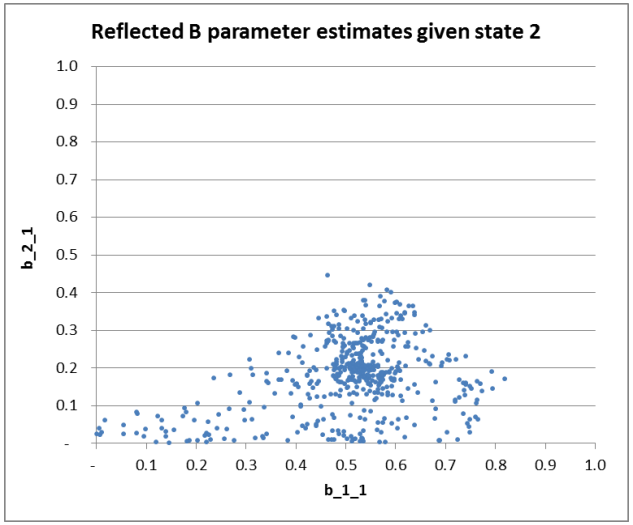
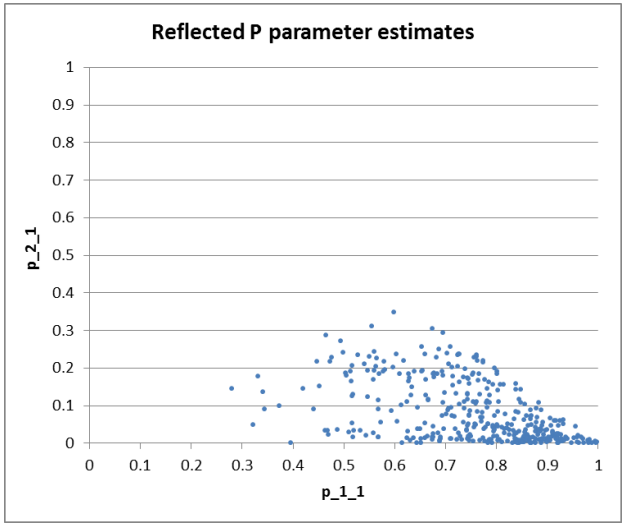
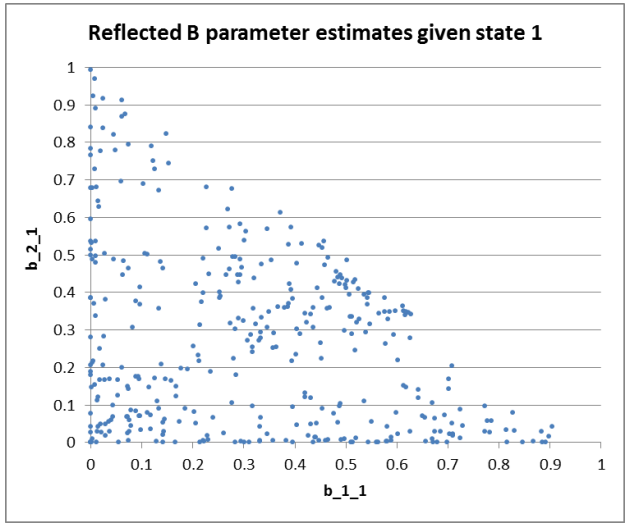


Figure 7.3.6: Scatter plot of the 500 BWA estimates contained in λ^* - using a signal sequence length of 2500.





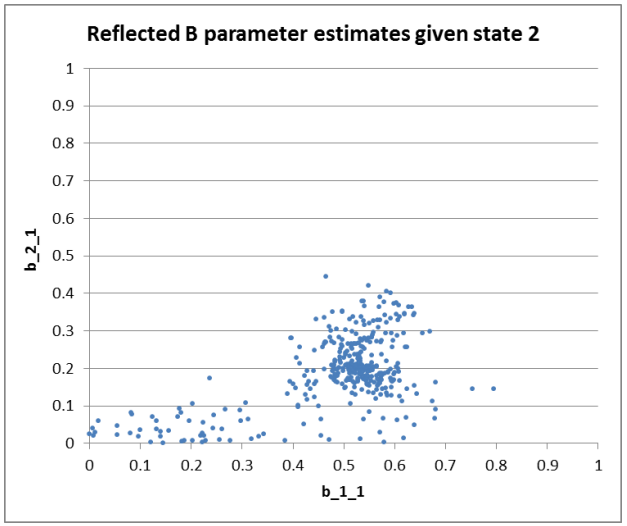
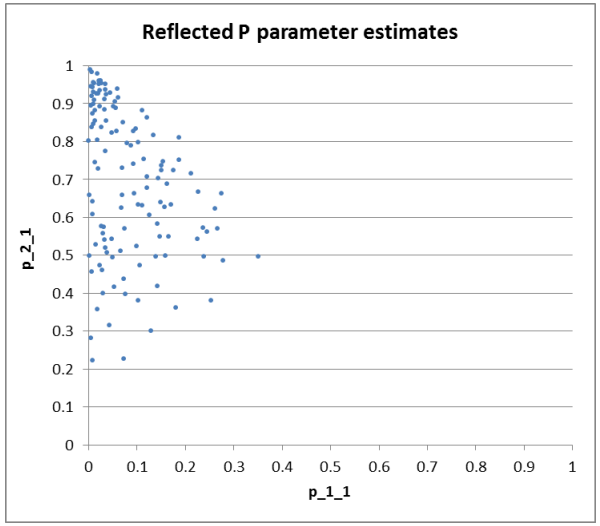
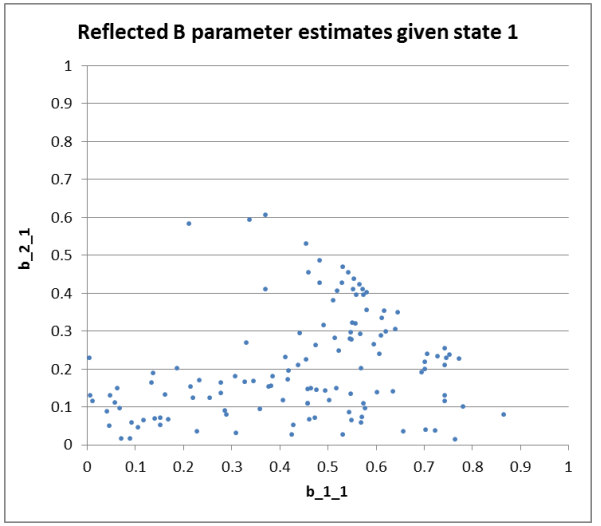


Figure 7.3.7: Scatter plot of the BWA estimates contained in λ^*, for which $\hat{p}_{11} > \hat{p}_{21}$ - using a signal sequence length of 2500.





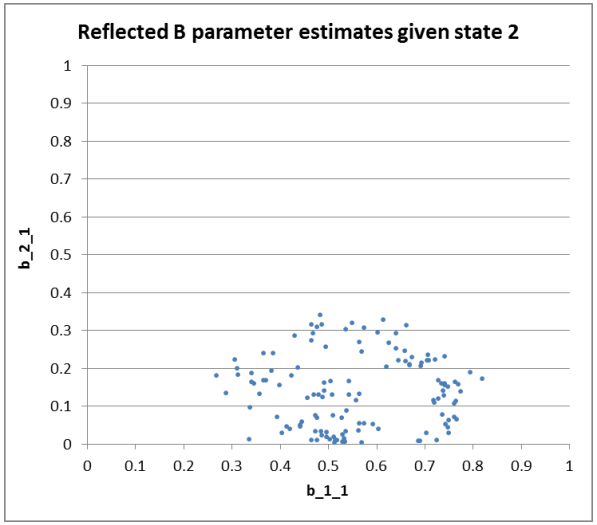


Figure 7.3.8: Scatter plot of the BWA estimates contained in λ^*, for which $\hat{p}_{11} < \hat{p}_{21}$ - using a signal sequence length of 2500.

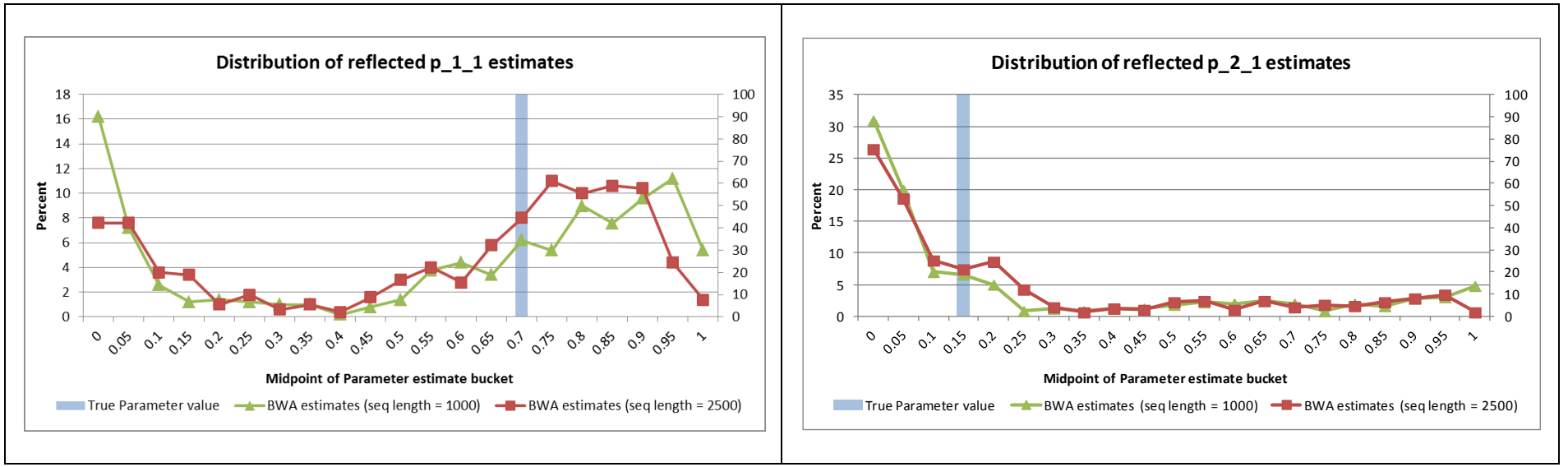


Figure 7.3.9: Frequency curves of the 500 BWA estimates (for the state transition probabilities) contained in λ^* - comparing a signal sequence length of 1000 and 2500.

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025, 0.025) to [0,975, 1.025)

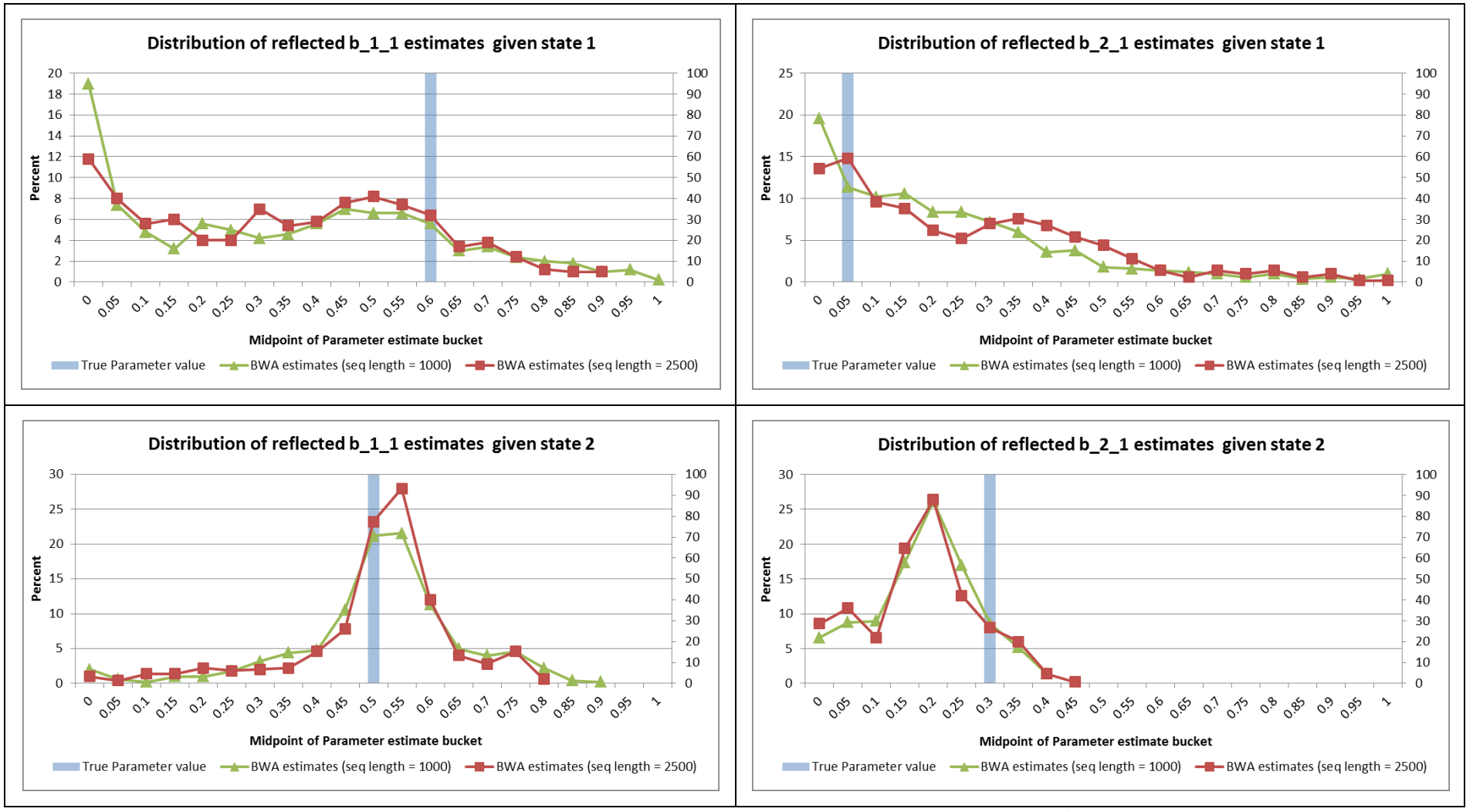


Figure 7.3.10: Frequency curves of the 500 BWA estimates (for the signal transition probabilities) contained in λ^* - comparing a signal sequence length of 1000 and 2500.

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025, 0.025) to [0,975, 1.025)

7.4 Exploring the Viterbi Algorithm for the DCMM

7.4.1 Simulation Results

The use of the Viterbi Algorithm (VA) to predict the underlying hidden state sequence of a DCMM was detailed in Section 6.2. This section will expand on the simulations described in Section 7.3 to explore how effectively the VA recovers the underlying hidden state sequence for the DCMM $\lambda = (\mathbf{P}, \mathbf{B}, \mathbf{a})$ defined in Section 7.3. It should be noted that due to the length of the signal sequences, scaling was needed to perform the VA. For this purpose scaling using the natural logarithm, as described in [37], was used. The scaling procedure is described for the HMM in [37], however it can also be extended to the DCMM, as was done in order to produce the results of this section.

To begin, recall from Section 7.3 that 500 separate state sequences (initially all of length 1000 time points) were simulated using the DCMM λ. For each simulated state sequence, a signal sequence was simulated. Each simulated signal sequence was then used to train the BWA to obtain an estimate of λ. Hence 500 separate BWA parameter set estimates were obtained (one set for each simulated signal sequence).

For each of the 500 simulated signal sequences, the VA was performed separately using both λ and the BWA estimate of λ associated with the signal sequence. The resulting Viterbi path was then compared to the simulated state sequence to measure the percentage of time points for which the state sequence was correctly predicted by the Viterbi path. These 500 percentages are shown in Figure 7.4.1, separately for when λ and the BWA estimate of λ was used to perform the VA. Let these distributions be denoted by P_1 and $\hat{P}_1$ respectively. In addition to this, there are two added plots in Figure 7.4.1. These were obtained by repeating the exercise described, but using a new set of 500 simulated state and signal sequences (while still using λ and the original BWA estimates of λ from Section 7.3) to perform the VA. Let these

distributions of the percentage of states correctly predicted be denoted by P_2 and $\hat{P}_2$ respectively. [4]

The above exercise was also repeated using the results from 7.3 when a sequence length of 2500 was used to perform the simulations. The distributions P_1, $\hat{P}_1$, P_2 and $\hat{P}_2$ corresponding to a sequence length of 2500 are shown in graph 7.4.2. In Figure 7.4.3 the two distributions for P_1, using a sequence length of 1000 and 2500 respectively, are plotted on the same graph; while Figure 7.4.4 shows the two distributions for $\hat{P}_1$, using a sequence length of 1000 and 2500 respectively. These four figures are discussed in the paragraphs below.

The distributions P_1 and P_2 are similar, as are the distributions $\hat{P}_1$ and $\hat{P}_2$. This is true for both a 1000 and 2500 sequence length. For this reason the remainder of discussion in this section will focus on P_1 and $\hat{P}_1$.

As expected, using a sequence length of 1000 or 2500 resulted in a similar distribution for P_1, with the percentage of states correctly predicted lying predominately between 62.5% and 72.5% for this particular DCMM. The distribution of $\hat{P}_1$ lies to the left of P_1 when either a sequence length of 1000 or 2500 is used. That is, using the BWA parameter estimates as opposed to the actual model parameters to perform the VA leads to the predicted Viterbi state path being less accurate (when compared to the actual simulated state path) - a somewhat expected result. What is also noticeable is that $\hat{P}_1$ appears to be bi-modal (for both a sequence length of 1000 and 2500). This is further explored in the next paragraph. Finally the distribution for $\hat{P}_1$ using a sequence length of 2500 lies to the right of the distribution for $\hat{P}_1$ when a sequence length of 1000 was used (see Figure 7.4.4). In Section 7.3 it was noted that using a sequence length of 2500, as opposed to 1000, to train the BWA produced more ac-

[4] In obtaining $\hat{P}_1$, the same simulated signal sequence which was used to train the BWA was also used (together with this BWA estimate) to perform the VA. This could lead to a potential bias in measuring the accuracy of the VA. This follows as using a BWA estimate in conjunction with the signal sequence used to train the BWA may produce a more accurate Viterbi path than if another signal sequence (not used to train the BWA) was used to perform the VA.

curate parameter estimates. Figure 7.4.4 then shows that using these more accurate BWA parameter estimates to perform the VA results in a more accurate Viterbi state path (when compared to the actual simulated state path).

The bi-modal nature of $\hat{P}_1$ is now explored. Recall from Section 7.3 that the 500 BWA estimates for p_{11} and p_{21} appeared to have two distinct groupings, one where $\hat{p}_{11} > \hat{p}_{21}$ and the other where $\hat{p}_{11} < \hat{p}_{21}$ (see for example Figure 7.3.6). The grouping $\hat{p}_{11} > \hat{p}_{21}$ represents more accurate estimates as the true $\mathbf{P}$ model parameters for λ are $(p_{11}, p_{21}) = (0.70, 0.15)$. As these BWA parameter estimates were used to perform the VA, this bi-modal nature of the BWA estimates could then result in the bi-modal nature of $\hat{P}_1$. Analysis shows that this is indeed the case. Figure 7.4.5 shows the sets of BWA estimates which resulted in the predication accuracy of the Viterbi path being between 47.5% and 57.5% (i.e. the left mode of $\hat{P}_1$ in Figure 7.4.2) when a sequence length of 2500 was used. Investigation of Figure 7.4.5 shows that most of these sets of BWA estimates are such that $\hat{p}_{11} < \hat{p}_{21}$. Hence the less accurate BWA parameter estimates have resulted in a less accurate Viterbi path.

It is also interesting to note the sets of BWA parameter estimates which resulted in a prediction accuracy of the Viterbi path less than 47.5%. These are plotted in Figure 7.4.6 for a signal length of 2500 and appear to be such that $(\hat{p}_{11}, \hat{p}_{21})$ is clustered around the point $(0.7, 0.2)$; $(\hat{b}_{11}^{(1)}, \hat{b}_{21}^{(1)})$ is close to the line $\hat{b}_{21}^{(1)} = \hat{b}_{11}^{(1)}$; and $\hat{b}_{21}^{(2)} < 0.15$. That is, for this particular DCMM λ, when BWA estimates in the vicinity of these regions are used to train the VA, the success rate of the Viterbi path in predicting the true state path is poor. Analysis revealed similar findings when a sequence length of 1000 was used.

Next the number of state transitions which occur in the state sequences is investigated. The results using a sequence length of 2500 are discussed below, however analysis revealed that similar comments hold when a sequence length of 1000 is used. To begin, the number of state transitions were determined for each of 500 simulated state sequences, the 500 Viterbi state paths when the actual model parameters were

used to perform the VA, and the 500 Viterbi state paths when the BWA parameter estimates were used to perform the VA. The mean of the number of transitions which occurred across the 500 sequences was then taken. In this way it was determined that the average number of transitions which occurred per simulated state sequence (of length 2500) was 499. Similarly it was determined that the average number of transitions which occurred per Viterbi state path when the actual model parameters were used to perform the VA was only 3.8. That is, considerably fewer transitions occurred within the Viterbi state paths when the actual model parameters were used to perform the VA than what occurred in the actual simulated state paths. To further explore this, the exercise was repeated for the alternative DCMMs specified by λ_1, λ_2 and λ_3 below. The average number of state transitions for the simulated state paths and the average number of state transitions for the Viterbi state paths (using actual model parameters to perform the VA) was 166 and 104 respectively for λ_1; 2,310 and 2,472 for λ_2; and 1,502 and 2,023 for λ_3. The percentage of states correctly predicted by the Viterbi path was also averaged over the 500 simulated sequences and found to be 93.1% for λ_1, 67.1% for λ_2, and 72.0% for λ_3. Medians for the above were also tested - these gave very similar results to the means.

Based on this analysis, it appears as if the accuracy of the VA (measured through the percentage of states correctly predicted by the Viterbi path and by comparing the number of state transitions in the Viterbi path and in the actual state sequence) can differ vastly depending on the parameters of the underlying DCMM. The specifications for the DCMMs λ_1, λ_2 and λ_3 discussed above are as follows:

$$\lambda_1 : \mathbf{a} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \mathbf{P} = \begin{pmatrix} 0.90 & 0.10 \\ 0.05 & 0.95 \end{pmatrix} \mathbf{B}^{(1)} = \begin{pmatrix} 0.20 & 0.80 \\ 0.90 & 0.10 \end{pmatrix} \mathbf{B}^{(2)} = \begin{pmatrix} 0.90 & 0.10 \\ 0.30 & 0.70 \end{pmatrix}$$

$$\lambda_2 : \mathbf{a} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \mathbf{P} = \begin{pmatrix} 0.05 & 0.95 \\ 0.90 & 0.10 \end{pmatrix} \mathbf{B}^{(1)} = \begin{pmatrix} 0.50 & 0.50 \\ 0.25 & 0.75 \end{pmatrix} \mathbf{B}^{(2)} = \begin{pmatrix} 0.30 & 0.70 \\ 0.50 & 0.50 \end{pmatrix}$$

$$\lambda_3 : \mathbf{a} = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} \mathbf{P} = \begin{pmatrix} 0.50 & 0.50 \\ 0.75 & 0.25 \end{pmatrix} \mathbf{B}^{(1)} = \begin{pmatrix} 0.20 & 0.80 \\ 0.60 & 0.40 \end{pmatrix} \mathbf{B}^{(2)} = \begin{pmatrix} 0.70 & 0.30 \\ 1 & 0 \end{pmatrix}.$$

Returning to the original DCMM λ, the average number of transitions which occurred per Viterbi state path when the BWA parameter estimates were used to perform the VA was explored. Analysis revealed that this average number of transitions was 605. Further investigation revealed however that a considerable number of the 500 individual Viterbi state paths had more than 2000 state transitions. This is high considering that the path length is 2500 (and that the average number of transitions which occurred per actual simulated state sequence was 499), but can be explained. Recall from Section 7.3 that of the 500 sets of BWA parameter estimates, there was a material number where $(\hat{p}_{11}, \hat{p}_{21})$ was close to the boundary point $(0, 1)$ - see for example Figure 7.3.6. A large number of state transitions would then be expected in the Viterbi state path when using these BWA parameter estimates to perform the VA. In fact, analysis showed that when the BWA parameter estimates with $\hat{p}_{11} < \hat{p}_{21}$ were used to perform the VA (this accounted for 127 of the 500 simulations) the average number of transitions which occurred per Viterbi state path was 2209. By contrast, when the BWA parameter estimates with $\hat{p}_{11} \geq \hat{p}_{21}$ were used to perform the VA (this accounted for 373 of the 500 simulations) the average number of transitions which occurred per Viterbi state path was 59. This number is considerably less than 499 - the average number of state transitions which occurred per simulated state sequence.

Finally the prediction accuracy of the Viterbi path for the DCMM of this section is compared to the prediction accuracy of the Viterbi path for the HMM used in Section 7.2. As has been previously mentioned, the two state, two signal HMM used in Section 7.2 is comparable to the two state, two signal DCMM used in this section. Figure 7.4.7 shows the following:

- The distribution of the percentage of states correctly predicted by the Viterbi path using the HMM from Section 7.2, a sequence length of 1000 and the actual model parameters to perform the VA. This is the solid blue line in Figure 7.4.7 and is equivalent to distribution P_1 of Figure 7.2.5.
- The distribution of the percentage of states correctly predicted by the Viterbi

path using the HMM from Section 7.2, a sequence length of 1000 and the BWA parameter estimates to perform the VA. This is the solid red line in Figure 7.4.7 and is equivalent to distribution $\hat{P}_1$ of Figure 7.2.5.

- The distribution of the percentage of states correctly predicted by the Viterbi path using the DCMM from this section, a sequence length of 1000 and the actual model parameters to perform the VA. This is the dotted green line in Figure 7.4.7 and is equivalent to distribution P_1 of Figure 7.4.1.
- The distribution of the percentage of states correctly predicted by the Viterbi path using the DCMM from this section, a sequence length of 1000 and the BWA parameter estimates to perform the VA. This is the dotted purple line in Figure 7.4.7 and is equivalent to distribution $\hat{P}_1$ of Figure 7.4.1.

From Figure 7.4.7 it can be seen that for this particular HMM and DCMM, the prediction accuracy of the Viterbi path is lower for the DCMM. This is true when either the true model parameters or the BWA estimates are used to perform the VA. In particular, for the HMM, the majority of the percentage of states correctly predicted by the VA lie between 77.5% and 87.5% when true model parameters are used to perform the VA, and between 72.5% and 87.5% when BWA parameter estimates are used to perform the VA. For the DCMM, the majority of the percentage of states correctly predicted by the VA lie between 62.5% and 72.5% when true model parameters are used to perform the VA, and between 47.5% and 72.5% when BWA parameter estimates are used to perform the VA. The distributions of the percentage of states correctly predicted by the VA for the DCMM lie notably to the left of the corresponding HMM distributions.

7.4.2 Concluding Remarks

In conclusion of Section 7.4, the results from various simulation exercises analysing the prediction accuracy of the VA have been discussed. These simulations have been per-

formed using the DCMM specified by λ. As these simulation exercises help quantify the expected accuracy of the Viterbi path for a given DCMM, they are recommended in applications of other DCMMs where the interpretation of the Viterbi path is an important objective of the analysis.

The following can be concluded for the DCMM λ from the analysis performed:

- The VA predicted the true underlying state sequence more accurately when the actual model parameters were used to perform the VA as opposed to the BWA estimated parameters. In particular for the majority of the simulation exercises (when sequences of length 2500 were used), the percentage of states correctly predicted by the VA ranged between 62.5% and 72.5% when the actual model parameters were used to perform the VA, and between 47.5% and 72.5% when the BWA estimated parameters were used to perform the VA.

- As the accuracy of the BWA parameter estimates used to perform the VA improved, the VA predicted the true underlying state sequence with more accuracy.

- The accuracy of the VA can differ vastly among different underlying DCMMs. This however was only tested using true model parameter values to train the VA.

- The number of state transitions were investigated. In particular it was found that, depending on the underlying DCMM, the number of state transitions predicted by the VA may differ vastly to the number of state transitions which actually occur in the true underlying state sequence.

- The VA predicted the true underlying state sequence considerably more accurately for the HMM than for the DCMM. This was tested using sequences of length 1000.

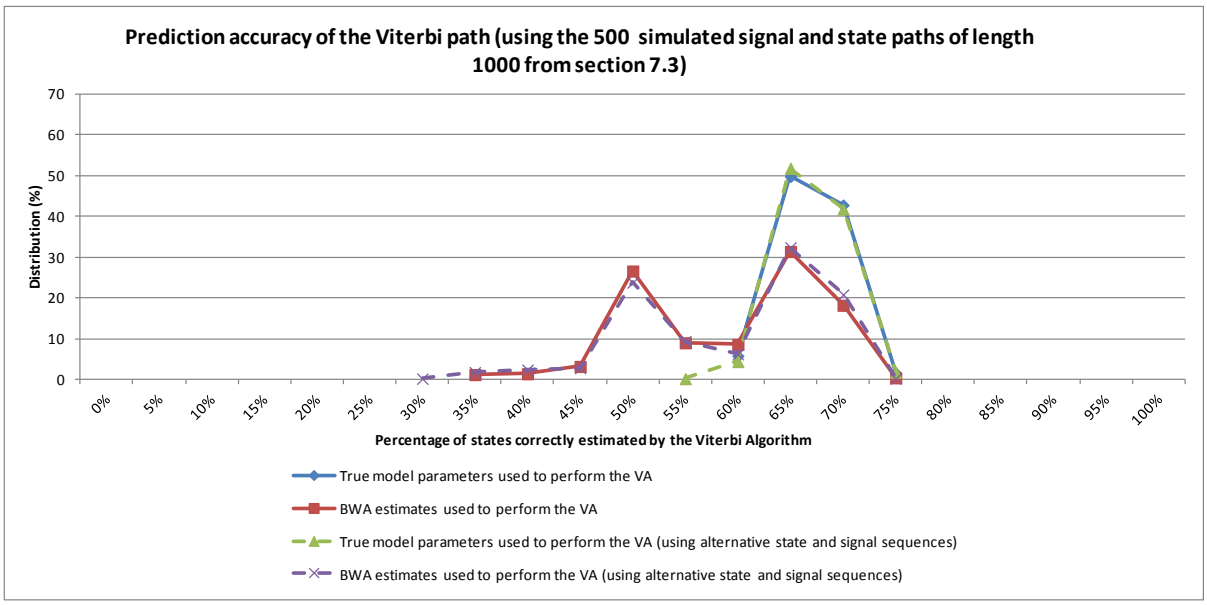


Figure 7.4.1: Prediction accuracy of the Viterbi path (using the 500 simulated signal and state paths of length 1000 from section 7.3).*

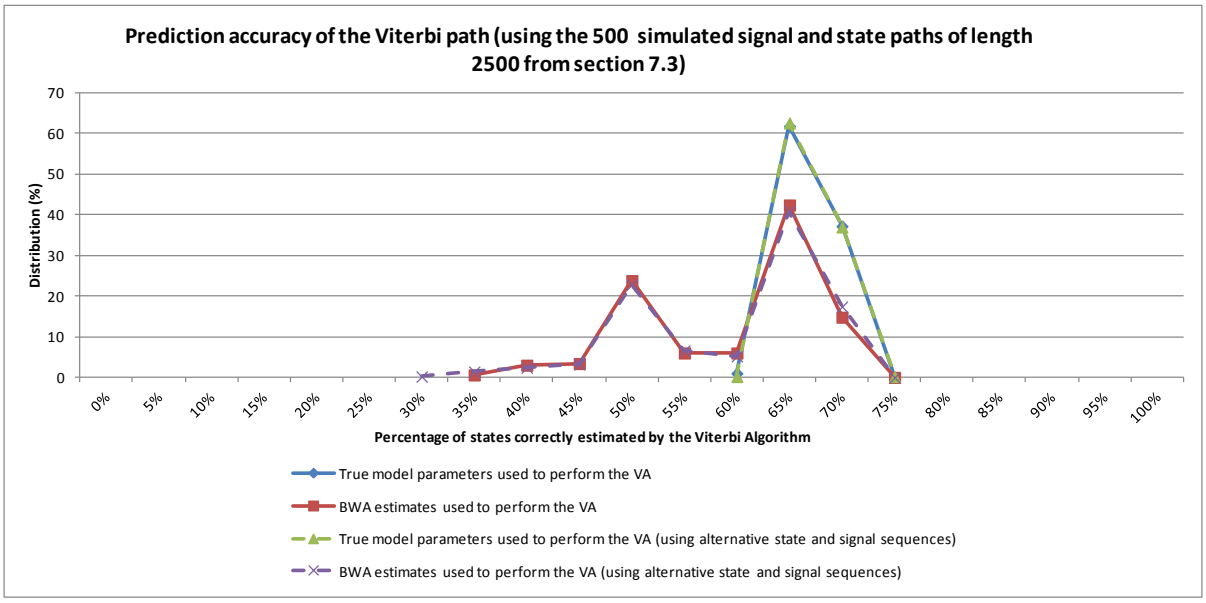


Figure 7.4.2: Prediction accuracy of the Viterbi path (using the 500 simulated signal and state paths of length 2500 from section 7.3).*

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

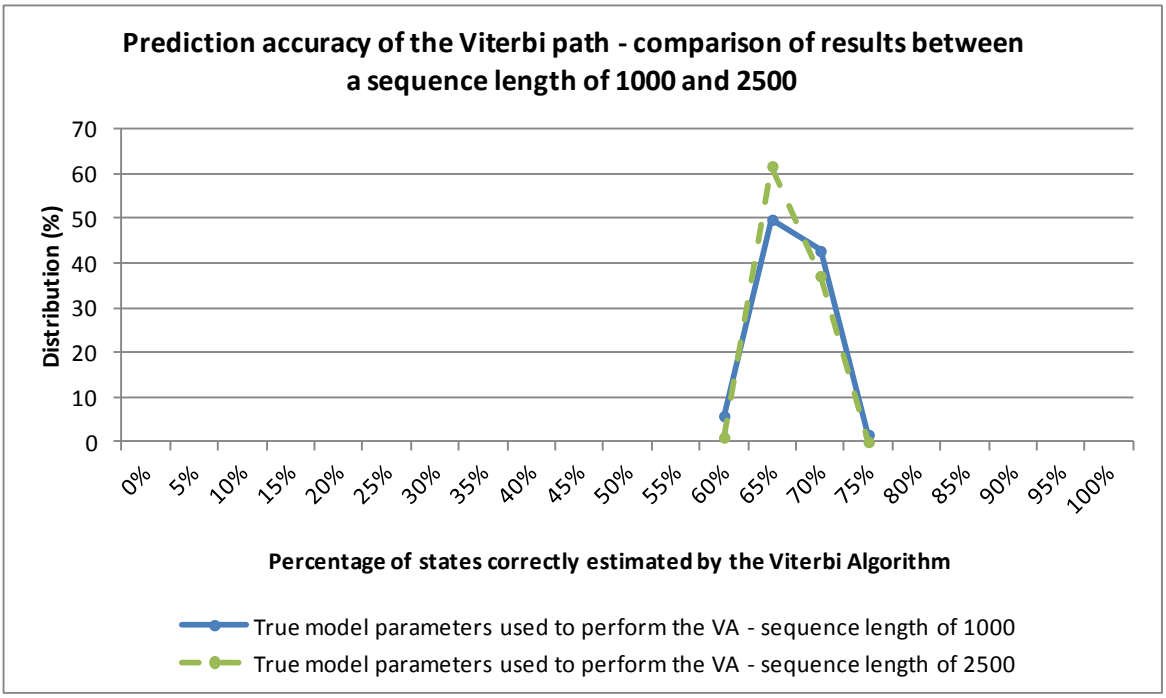


Figure 7.4.3: Comparison of the prediction accuracy of the Viterbi path when a sequence length of 1000 and a sequence length of 2500 are used. The actual model parameters were used to perform the Viterbi algorithm.*

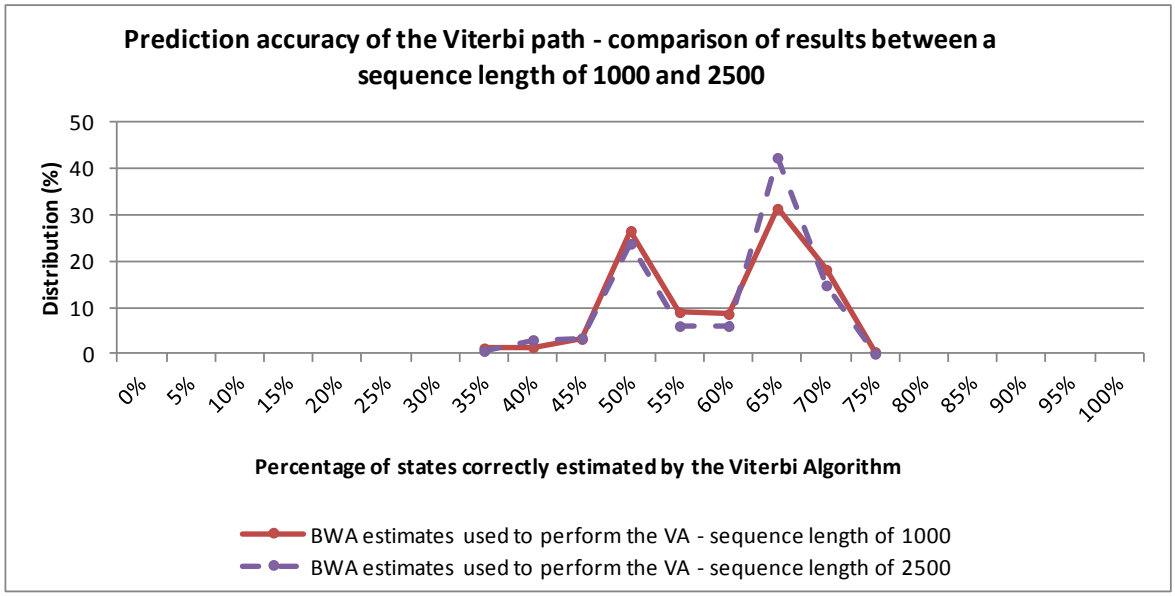
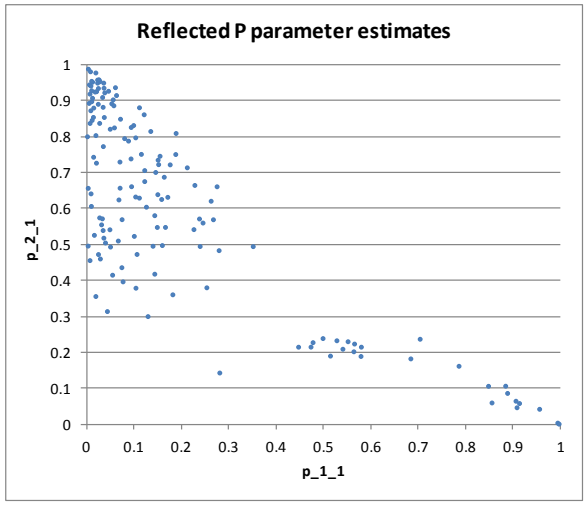
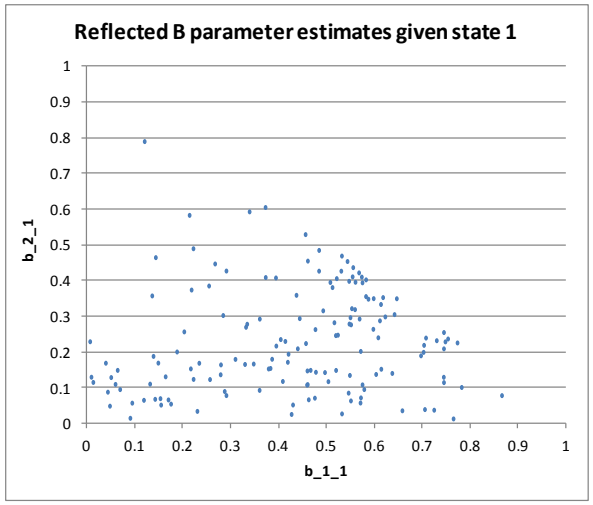


Figure 7.4.4: Comparison of the prediction accuracy of the Viterbi path when a sequence length of 1000 and a sequence length of 2500 are used. The estimated BWA model parameters were used to perform the Viterbi algorithm.*

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)





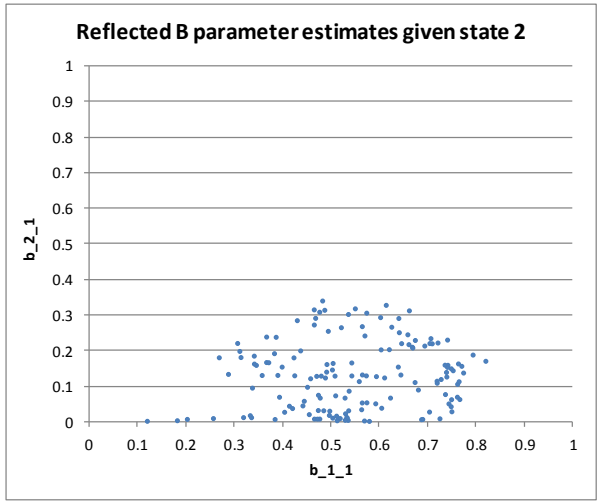
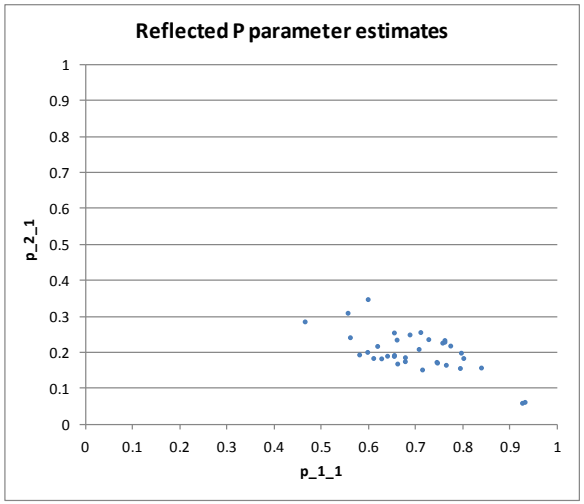
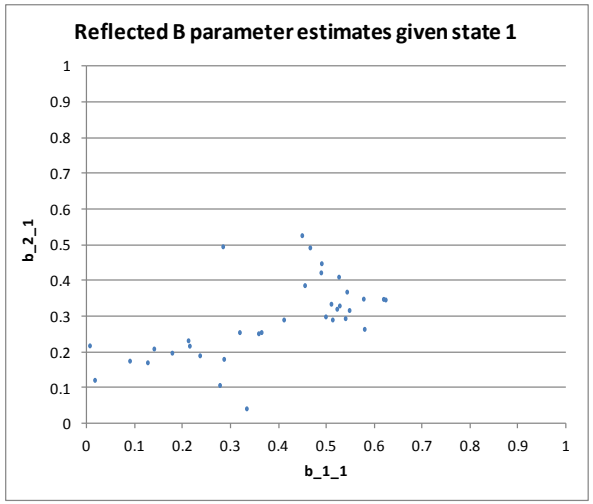


Figure 7.4.5: The BWA parameter estimates which, when used to perform the Viterbi Algorithm, led to a poor prediction accuracy of the Viterbi path of between 47.5% and 57.5%





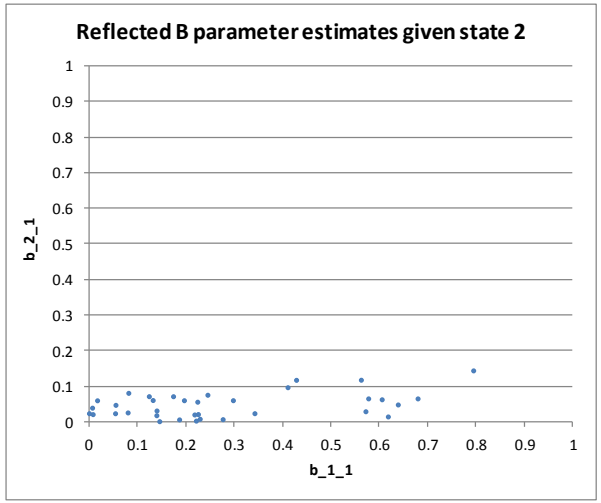


Figure 7.4.6: The BWA parameter estimates which, when used to perform the Viterbi Algorithm, led to a poor prediction accuracy of the Viterbi path of less than 47.5%.

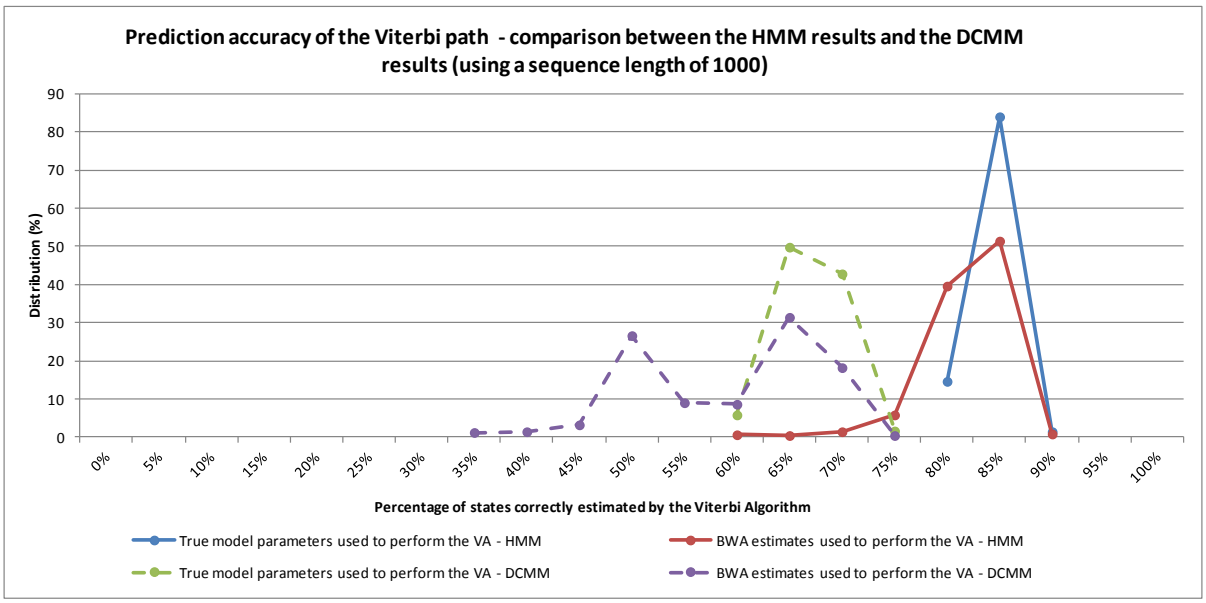


Figure 7.4.7: Prediction accuracy of the Viterbi path – a comparison between the HMM and the DCMM when a sequence length of 1000 is used.*

*Note that values on the x-axis represent the midpoint of the bins. The bins range from [-0.025,0.025) to [0.975,1.025)

7.5 Additional Simulation Studies

The simulation exercises presented in this chapter have provided insights into the mechanics, the effectiveness and the shortcomings of the BWA and the VA for both HMMs and DCMMs. Scope for simulations within the HMM and DCMM framework is vast, and hence many other simulation studies may also be explored. Review of the literature reveals numerous simulation exercises which have been performed by various authors, particularly for the HMM.

For example [15] uses simulated data to compare the performance of three different parameter estimation approaches for the HMM - namely numerical maximisation of the log-likelihood function (discussed in Sections 4.2 and 4.3 of this dissertation), the EM algorithm (which is aligned to a BWA approach) and a hybrid approach proposed by [15]. Similar to the simulation study of this dissertation, the properties of these different approaches are investigated (for example the dependence on starting values, the influence of different signal sequence lengths and the stability of estimation). Bootstrap-based confidence intervals are also explored in [15].

In [18] HMM simulations are used to compare different parameter estimation approaches (all based on a Baum-Welch approach) when multiple signal sequences are observed. Mention is made in [18] of the possibilities of 'local minima traps' when performing the BWA. This was also noted in the simulation study presented in this chapter.

While not as numerous as for the HMM, simulation studies for the DCMM are also discussed in the literature. One interesting example is presented in [10]. In [10] a three-state two-signal DCMM is used to simulate 20 signal sequences each of length 504. Using the simulated data, 11 different models were fit to each sequence: the independence model, Markov chains of order 1 to 4, HMMs with 2, 3 and 4 hidden states and DCMMs with 2, 3 and 4 hidden states. HMMs and DCMMs were fit using the BWA. Models were classified according to their BIC value and in all but one of

the 20 cases, the best model was a DCMM. It was thus concluded in [10] that the DCMM can represent non-homogeneous time-series better than homogeneous Markov chains and HMMs.

An interesting observation from [10] is that for most of the 20 signal sequences, the fitted two-state DCMM ranked better than the fitted three-state DCMM (according to the BIC value) - despite the fact that the signal sequences were simulated from a three-state DCMM. A possible reason given in the paper is that the second state (of the DCMM used to create the simulated sequences) represents independence, that is $b_{11}^{(2)} = b_{12}^{(2)} = b_{21}^{(2)} = b_{22}^{(2)} = 0.5$. The paper thus makes the observation that 'it appears that the model concentrates upon the informative part of the data and does not add extra parameters for the non-informative independence situation'.

Mention is made in [10] that when interpreting the estimated BWA parameters, reflection of the estimates may be required in order to make comparisons to the true model parameters of the original DCMM. This was also seen and noted in the simulation study presented in this chapter.

Finally signal sequences of length 504 were simulated for the study presented in [10]. This seems low based on the results and findings of the simulation study presented in this chapter. However a contributing factor may be the choice of the DCMM used to generate the simulations. In [10], probabilities within $\mathbf{P}$ are greater than 0.8 on the diagonal, less than 0.1 otherwise, and such that it is not possible to go in one step from state one to three and vice versa. This will no doubt limit the number of transitions within the hidden state process. All probabilities within $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(3)}$ are either greater than 0.9 or less than 0.1. Such a DCMM may require less data being needed to accurately perform the BWA. This is an area of future study which can be addressed through simulations similar to those presented in this chapter.

Chapter 8

Selected HMM and DCMM Applications

In order to appropriately conclude the discussion provided in this dissertation, an overview of selected HMM and DCMM applications in the literature is provided in Sections 8.1 and 8.2. Section 8.3 provides a focussed review of HMM and DCMM applications within the field of credit risk. The simulation studies performed in Chapter 7 of this dissertation will also be referenced to determine if conclusions from this study can be linked to the applications presented in the literature.

8.1 Selected HMM Applications

A review of the literature reveals that HMMs have been applied to numerous fields. These include applications in the field of *pattern recognition* tasks such as speech recognition, face recognition, handwriting recognition, gesture recognition, human identification using gait and facial expression identification from videos (see for example [31], [37] and [43]), *credit risk* (details provided later in this section), *biology* (for instance gene prediction and the study of DNA and protein sequences, see for example [29]) and *partial discharge* (the study of localized electric breakdowns of small portions of solid or liquid electrical insulation systems under high voltage stress, see

for example [42]).

Focusing on recognition tasks, the usefulness of the HMM stems from its ability to learn HMM parameters from observation sequences (through for example the Baum-Welch re-estimation procedure), and then consequently its ability to assess the likelihood that a new observation sequence is associated with a learned HMM (using for example the evaluation procedure described in Section 3.1). For instance, consider speech recognition - a feature of most modern cellular phones (voice dialling), bluetooth kits, vehicle navigation systems, and used extensively among people with disabilities to their hands who require alternative means of input into a computer. The use of the HMM within speech recognition is summarised as follows (numerous references within the literature, for example [31] and [37], may be consulted should additional details be required).

Assume a vocabulary of J words denoted by $V = \{w_1, w_2, ..., w_J\}$. Now,

1) A training set is collected for each word $w_i \in V$ by speaking w_i into a signal processor M_i times, where each time w_i is spoken it is converted into an observation sequence. In other words, w_i will have a training set consisting of M_i observation sequences.

2) For each $w_i \in V$, the training set which has been obtained for w_i is used to train a distinct HMM for that word, denoted λ_i. This can be achieved through some parameter estimation procedure, for the example the Baum-Welch algorithm.

3) Define $\tilde{w}$ to be the unknown input word that is spoken and is to be recognized by the speech recogniser.

Also, define w^* to be the word in V that the speech recogniser identifies $\tilde{w}$ to be.

w^* is then selected from V as follows:

- Firstly $\tilde{w}$ must be converted into an observation sequence, $\mathbf{O}$, by the signal processor.
- $P(\mathbf{O}|\lambda_i)$ is calculated for each $i = 1, 2, ..., J$, using an evaluation procedure - for example the evaluation procedure discussed in Section 3.1 of this dissertation.
- $w^* = \arg \max_{1 \leq i \leq J} \{P(\mathbf{O}|\lambda_i)\}$ will then be given as the recognized word, provided that $\max_{1 \leq i \leq J} \{P(\mathbf{O}|\lambda_i)\}$ is greater than some pre-defined minimum probability level. If this is not the case, a default message such as "No match found" will be given as the output instead of w^*.

Further flavour of the range of HMM applications can be obtained from [35] and [46]. In these publications the HMM framework is used in a variety of studies including modelling epileptic seizure counts, births at the Edenvale hospital, homicides and suicides in Cape Town, wind direction at Koeberg, and the trading of shares on the Johannesburg Stock Exchange (JSE). A selection of recent applications of the HMM framework include [2], [14], [16], [26], [28] and [45]. These papers are overviewed in the paragraphs below.

A HMM approach is used in [26] to model the availability or expected life of electrical, electronic and electromechanical systems and products. In [2] a HMM is used to analyse irrigation decision behaviour of farmers and make forecasts of their future decisions. The motivation for this study is that canal operators will typically divert water from rivers to a field after receiving a water order from a farmer. Hence if farmers' irrigation decisions could be better anticipated, it would be possible to improve canal operations using improved future water demand estimates. In this study, irrigation decisions were represented by the hidden states of the HMM and were estimated using the Viterbi Algorithm (VA), discussed in Section 3.2 of this dissertation. In [14] the streamflow of the Upper Colorado River Basin is modelled using a gamma HMM. According to [14], hydroclimate time series (the underlying driver of streamflow) often exhibit low year-to-year autocorrelation while showing prolonged

wet and dry periods reminiscent of regime-shifting behaviour. A HMM framework is then well suited. In this particular study, the hidden states are described as climate regimes and the observed sequence, dependent on the current state, is the streamflow measurements. The gamma distribution is commonly used in hydrologic modelling (e.g. streamflow and rainfall time series) because of its lower bound of zero. And so a gamma distribution is used for each hidden state of the HMM to model the streamflow. A deviation from the classical gamma HMM is also presented in [14]. It is noted that state transitions are likely to exhibit non-stationarity and is addressed in [14] as follows. The classical gamma HMM is first assumed and parameters are estimated using an EM algorithm approach; these estimated parameters are then used to perform the VA and decode the hidden states. This decoded state sequence is used to train a multinomial logistic regression model (where climate indices are used as explanatory variables) to obtain estimates of the probability that the system was in a given state at a given time point.

A 3-state Poisson HMM is used in [16] to forecast the expected annual frequency of earthquakes (until the year 2047) with magnitudes greater than or equal to 4 in the Bilecik region in Turkey. This interpretation of the HMM is similar to that which was discussed in Section 2.2 of this dissertation. Analysis presented in [16] compares the expected number of earthquakes using the Poisson HMM fit to the data to the expected number of earthquakes using a homogeneous Poisson process fit to the data. Comparison to the actual frequency of earthquakes observed in the 113 years of historical data reveals that the Poisson HMM predicts materially more accurately than the homogeneous Poisson process. The study presented in [45] proposes using a Gaussian HMM to analyse drought patterns in South Korea and the role that typhoons play in ending drought conditions. In this study the state sequence of the HMM represents the latent weather state [1] and observed monthly rainfall amounts are modelled using a normal distribution dependent on the state of the HMM. Estimating and analysing

[1]Seven states are assumed which represent weather conditions from extreme drought (state 1) to extreme wet conditions (state 7).

the hidden state path then allows drought patterns to be studied, and the beginnings and endings of drought periods to be classified. Several aspects are studied in [45], for example the evolution between drought and wet conditions, the role which typhoons play in ending drought conditions, and what precipitation conditions are expected for the months following a typhoon. It is noted in [45] that the advantage of the HMM framework over other drought analysis tools is that "the HMM explicitly takes into account the temporal dependence in the drought states so that smoothed transition probability between drought states over time is clearly identified, while the SPI [2] showed a sudden change in the transition of drought or wet states".

The use of HMMs in autonomous vehicles to correctly detect the state of a traffic light (red, yellow, green or no detection [3]) is investigated in [28]. According to [28] multiple authors have used image processing as a base for achieving traffic light detection. To achieve this images captured by a camera located on the autonomous vehicle are processed to detect traffic lights and determine the active state of the traffic light. However, adverse lighting and/or weather conditions can result in either the state of an identified traffic light not being detected or the state of an identified traffic light being incorrectly detected. It is proposed in [28] that the HMM be used to improve the detection of the active state of a traffic light as follows. As the autonomous vehicle approaches the traffic light, several images are captured and processed, and the state of the traffic light is detected for each image. This then forms the signal (output) sequence for the HMM, with signal space {red, yellow, green, no detection}. Now, the true sequence of active states of the traffic light will possess the Markov property. This sequence can then be represented by the hidden state sequence of the HMM. Hence the state space of the HMM is also {red, yellow, green, no detection}. The VA is then used to estimate the true sequence of active states of the traffic light from the sequence of traffic light states detected from the image processing (the sig-

[2] [45] notes that the standardized precipitation index (SPI) is the most commonly used drought index.

[3] For example, the traffic light is not working or there is no traffic light present.

nal sequence). If the VA performs accurately then the predicted Viterbi path will accurately capture the true sequence of active states of the traffic light even if errors in detecting the state of the traffic light during image processing occurred. A study presented in [28] resulted in the proposed HMM approach obtaining 90.55% accuracy in the detection of the traffic light state, versus 78.54% accuracy obtained using solely image processing.

8.2 Selected DCMM Applications

While fewer, there are several examples in the literature of studies where the DCMM has been applied - see for example [10], [11], [22], [23] and [46]. These include using a DCMM to model credit rating transitions ([22], [23]), wind speeds in order to determine the feasibility of wind power ([10]), births at the Edenvale hospital ([46]), DNA analysis, behaviour of young monkeys and the phrases of a bird call/song ([11]).

The application presented in [10] is discussed next. Recall that this study was also mentioned in Section 6.1.1 of this dissertation to illustrate the differences between Markov chains, HMMs and DCMMs. In this study the average daily wind speed during the period 1961-1978 (a time series of length 6574 data points) was analysed in order to determine the possibility of wind power. As exceptionally low and high wind speeds can prevent good exploitation of this power, the data was classified into three categories: low wind speed, normal wind speed and high wind speed. Now intuitively it would be expected that the wind speed on a given day is correlated with its speed the previous day, but that the process is not stationary and evolves throughout the year as the seasons change. Hence a DCMM seems well suited to represent this data. However several other models were also fit to the data (including the independence model, Markov chains of varying orders, HMMs of varying orders and mixture transition distribution (MTD) models). According to the BIC values, the two-state DCMM was identified as the most appropriate model. After analysing

the estimated model parameters, the first hidden state was interpreted in [10] to be a situation of low wind speeds and the second hidden state was interpreted to be a situation of high wind speeds.

Linking the above mentioned study presented in [10] to the simulation studies of this dissertation (Chapter 7), the following can be noted. Firstly, the BWA was used by [10] to estimate the DCMM model parameters. This is consistent to the approach used in Chapter 7 of this dissertation. In these simulation studies it was recommended that in practice, in order to perform the BWA and fit an unknown DCMM, a signal sequence longer than 2500 data points be used. This was based on analysis of the sampling distributions of the BWA parameter estimates. Encouragingly the signal sequence used in [10] consisted of 6574 data points. In the simulation exercises in Chapter 7 of this dissertation, it was also noted that starting parameter values used to train the BWA can greatly affect the final estimated parameters (as the BWA finds a local maxima of the likelihood function rather than a global maxima). In these simulation exercises, 150 starting parameter sets were randomly created (such that the required DCMM probability properties held) and used to train the BWA - the final BWA estimate set which was selected was that which yielded the highest likelihood value. In [10], no mention is made regarding the selection of the starting parameter values used to train the BWA. Encouragingly however, while discussing the theory of the BWA, the following comment is made in [10]: "Since we cannot insure that this procedure converges to the global maximum of the likelihood rather than to a local maximum, the choice of starting values is critical". Finally, as mentioned above, the first hidden state was interpreted in [10] to be a situation of low wind speeds and the second hidden state was interpreted to be a situation of high wind speeds. It would indeed be interesting to further this study and use the estimated DCMM model presented in [10] to perform the Viterbi algorithm and estimate the hidden state path. In this way it could be seen if, for example, the periods when the hidden path was estimated to be in state 2 corresponded to typically windy months/seasons.

The premise of using BWA parameter estimates to perform the Viterbi algorithm and estimate the hidden state path was simulated and investigated in Section 7.4 of this dissertation.

8.3 HMM and DCMM Applications Within the Field of Credit Risk

The final HMM and DCMM applications which will be discussed are the use of these models within a credit risk framework. From a banking sector point of view, a bank would view credit risk as the risk of default on a debt that may arise from a borrower failing to make required payments. Papers in which the HMM has been applied in a credit risk context include [3], [4], [27], [30]; and papers in which the DCMM has been applied in a credit risk context include [22] and [23].

The first application which will be discussed is the HMM framework used in [3], [4] and [27]. In these papers the occurrence of defaults within a portfolio of corporate bonds is modelled as a hidden Markov process. In particular, the hidden state process is assumed to represent the state of risk within a sector. The signal or observed process of the HMM is the number of defaults which occurred within the portfolio at each time point, whereby it is assumed that the number of defaults is conditionally dependent on the hidden state via the binomial distribution. This is then an example of the distribution HMM which was discussed in Section 2.2 of this dissertation. The analysis of interest in this application framework is around the hidden state sequence as estimation of the hidden state path allows for the detection of periods of enhanced risk in the credit cycle.

The study presented in [27] assumes that the hidden state can take one of two values: 0 (representing an underlying state of normal risk) and 1 (representing an underlying state of enhanced risk). This is extended in [3] and [4]. In particular the work

presented in [4] no longer assumes only two hidden states[4]. In addition the state transition probabilities are not assumed to be time homogeneous. Instead now the state transition probabilities are modelled as a function of observed covariates (for example macroeconomic variables). This is done through regression using a logistic link function. It is shown in [4] that making use of covariates to predict the state transition periods enables sharper identification of periods of high and low default regimes. In the study presented in [3], the performance of the HMM application is examined using different specifications of the hidden state space. In particular both discrete-state and continuous-state HMM specifications are considered. The effect of mis-specification of the hidden layer is also investigated. Regarding this, it is stated in [3] that "this appears particularly important given the limited number of time series observations typically available for default modelling: annual, quarterly, or monthly time series since the 1980s." Based on the simulations performed in Section 7.1 of this dissertation (where it was discovered that if the observation sequence used to train the BWA is not sufficiently long enough, the BWA estimates are likely to contain material variance and some degree of bias), this statement in [3] seems valid. In particular the observation sequence used to train the BWA in [27] consisted of 88 time points. Based on the simulations performed in Section 7.1 of this dissertation (all be it not for the binomial HMM used in [27]), a sequence of this length could result in inaccurate estimates if a mis-specified HMM is used. This is further discussed in the paragraphs below.

The study performed in [27] is now examined in more detail. Model parameters are estimated in [27] using the BWA adapted for the binomial distribution HMM[5]. These estimated parameters are then used as input to perform the VA to estimate the hidden state path, thereby analysing which periods in history corresponded to a state of normal risk and which corresponded to a state of enhanced risk. Interestingly, when bonds were divided into their appropriate industry sectors and the sectors modelled

[4] In [4] a discrete state space is assumed with arbitrary s possible states.

[5] Adaptation of the BWA for distribution HMMs was discussed in Section 4.1.4 of this dissertation.

separately (four sectors were considered - Consumer, Energy, Media and Transport), the periods where the hidden state represented enhanced risk differed between sectors, although correlation could be observed. That is, while correlated to some degree, the credit risk cycles did show signs of differing between the different industry sectors. Knowing where in the credit cycle a particular industry sector is can be of great value to a risk manager. This is particularly true as detection of an enhanced credit risk state can serve as an early warning mechanism for high default regimes.

As an additional study, data from the different industry sectors was aggregated and the modelling re-performed (this was done for US issuers only). Hence the enhanced risk hidden state can now be seen as being related to global economic factors, affecting all sectors at the same time. Interestingly, since 1990, there have been two recessions in the US economy and periods where the hidden state represented enhanced risk overlapped with these two periods of recession. In both cases, the enhanced risk state anticipated the onset of recession and continued for a few months after the recession had ended. This indicates that the hidden state process can be used to detect enhanced risk in the credit cycle before the economy moves into recession, and also indicates that the credit cycle remains at higher risk for a few months after the recession has passed. This knowledge can be of great value to a risk manager.

In summary, [27] used a HMM approach to model the hidden layer present in default rate dynamics. This hidden layer can be viewed as the state of the credit cycle, and thus has an influence on the number of defaults expected in a portfolio. It is also shown in [27] that the economic cycle does not fully explain the credit cycle (hence the need to model this hidden layer).

Linking the studies of [27] to the theory and simulation exercises discussed in this dissertation, the following can be noted.

- The observed data used for this study was the number of defaults, measured in quarterly intervals over the period Q1 1981 to Q4 2002. This gives rise to

an observed sequence of length 88 time points. In order to determine if this sequence is sufficiently long enough, the sampling distribution for the BWA estimates can be determined through simulations (as was done in Section 7.1 of this dissertation). A similar simulation was indeed performed in [27] and the sampling distributions for the $\mathbf{P}$ parameters are comparable (the other model parameters could not be compared as a binomial distribution was not used to output the signals in Section 7.1). Interestingly, in Section 7.1 of this dissertation the presence of outlying BWA estimates in the sampling distributions for the $\mathbf{P}$ parameters was noted. This was also noted in [27].

- The primary focus of the application study in [27] is to analyse the hidden state sequence. The effectiveness of the Viterbi algorithm to retrieve the hidden state path for the HMM in question is investigated in [27] through simulation. The Viterbi path appears to agree remarkably well with the true state path. However this simulation only considered a single simulated state and observation sequence. This study could be furthered by considering multiple simulated sequences and obtaining a distribution showing how well the Viterbi path estimated the true state path across the various simulated sequences, as was considered in Section 7.2 of this dissertation.

- As has been mentioned, the hidden state sequence gives potentially valuable insights into the state of risk in the credit cycle. Being able to accurately forecast the hidden state for future time points would no doubt also be valuable. A further study of interest would then be, through simulations, to determine how accurately the hidden state sequence can be foretasted for future time points. An approach to forecasting the hidden state sequence for future time points was given in Section 3.4.3 of this dissertation (see equation (3.19)).

Not mentioned in [27] is whether the proposed HMM is suitable for predicting the number of defaults for future time points (recall that the primary focus of the study

was to analyse the hidden state path to determine the state of risk in the credit cycle). This however would be questionable, as it is assumed that each loan within the portfolio has equal probability of defaulting (the probability parameter of the binomial distribution used to model the number of defaults). That is, the credit assessment of an individual loan would not be taken into account when assessing its probability of defaulting. Furthermore, by construction of the model, the expected portfolio default rate (total number of defaults divided by total number of loans in the portfolio) would remain constant while the process remains in the same hidden state.

Another application area (within credit risk) of HMMs and DCMMs discussed in the literature concerns credit ratings, as detailed in [22], [23] and [30]. It is common practice in the banking industry to assign a credit rating to a borrower in order to determine the credit quality / credit worthiness of the borrower. Typically these credit ratings are either performed by external rating agencies (for example Moody's) or internally by the bank itself. Reviews of the credit rating are performed periodically, and a rating change signifies improvement (upgrade) or deterioration (downgrade) in a borrower's credit worthiness. Hence these dynamics are typically summarised in a transition matrix, where each entry in the matrix represents a probability of a credit rating migration. Furthermore it is standard practice to consider a Markov chain representation of credit rating dynamics.

It is claimed by [30] that published credit ratings may not always accurately reflect 'true' credit worthiness due to the fact that the posted credit ratings may sometimes be 'noisy or incomplete'. This is motivated in [30]. It is proposed in [30] that a HMM can better reveal 'true' credit worthiness. In particular it is supposed that the 'true' credit quality evolution is described by the hidden state Markov chain of the HMM; and that the published credit rating is the signal/observation sequence of the HMM. As a portfolio will typically consist of several borrowers, several observation sequences will be available to train the HMM and estimate the model parameters (Section 4.1.3 of this dissertation discussed adapting the BWA to cater for multiple observation

sequences). Once the HMM parameters have been estimated, $\hat{\mathbf{P}}$ then represents the 'true' credit rating transition probabilities and $\hat{\mathbf{B}}$ represents the probabilities that the published credit ratings are different from the 'true' credit rating. Of course if $\hat{\mathbf{B}}$ is estimated to be the identity matrix (or something close to it) then this implies that the published credit ratings represents the 'true' credit quality of a borrower. The application presented in [30] found this not to be the case. It is therefore suggested in [30] that, due to 'noise and incompleteness', the published credit ratings do not always represent the 'true' credit worthiness of a given borrower. Instead analysis of $\hat{\mathbf{P}}$ and $\hat{\mathbf{B}}$ can yield a better understanding of the 'true' credit quality.

Interestingly, there is alignment of the HMM interpretation between [30] (within credit risk) and [28] (recall that this paper was detailed in Section 8.1 and discusses applying a HMM to improve traffic light detection in autonomous cars). In both papers it is assumed there may be 'noise' present in the data observed, but that the hidden state process of the HMM is the true representation (of credit worthiness in [30] and the traffic light state in [28]).

A DCMM application is used in [22] and [23] to model the credit rating migration dynamics of a portfolio. This is indeed of interest in the banking industry as a rating migration of a company, to which the bank has loaned money to, signifies a change in the likelihood that the company may default on its loan.

Credit rating transition probabilities are commonly estimated in practice using a discrete time, time homogeneous Markov chain. However, as discussed earlier in this section, studies presented in [3], [4] and [27] suggest that there are also so-called hidden factors or risks driving the credit cycle[6]. This is likely to influence the evolution of credit ratings over time, which could result in non-stationary behaviour. This however is unlikely to be catered for by the discrete time, time homogeneous Markov chain. The work presented in [22] and [23] caters for this through the use of

[6]Furthermore the studies presented in [3], [4] and [27] suggest that these hidden risks depend on each other in successive periods - i.e. possess the Markov property.

the DCMM[7]. The hidden state process is taken to be the state of risk in the credit cycle (which is assumed to follow a Markov process). The observable process (which is also assumed to follow a Markov process dependent on the state occupied) is the credit rating which a given loan receives over time. In this way, the state of risk in the credit cycle, together with the Markov process describing the rating migrations, will determine the credit ratings of the loan through time. More detail on both [22] and [23] is given below.

To begin, it is acknowledged in [22] that the presence of rating drift[8] is noted in the literature. This is catered for in [22] in the DCMM framework by considering higher orders in both the state and signal Markov chains. The data used for the study in [22] consisted of 11,284 rated companies over 11 years of rating history (however, depending when a company might have opened or closed, rating information will not be available for 11 years for each company). As each rated company will give rise to an observation / signal sequence with which to train the DCMM, multiple observation sequences are available to estimate model parameters[9]. This gives rise (according to [22]) to almost 48,000 rating observations with which to train the DCMM. This is well in excess of the 2,500 observations which were used in the simulation study in Section 7.3 of this dissertation (this is encouraging as the simulation study showed material bias and variance in some of the BWA parameter estimates when 2,500 observations were used).

Various models were fit to this observed data. These included the Independence model, homogeneous Markov chains of different orders, MTD models of different orders, different combinations of HMMs (varying numbers of hidden states and varying orders in the hidden states) and different combinations of DCMMs (varying numbers of hidden states, varying orders in the hidden states, and varying orders in the

[7]A discrete-time, discrete-state space and discrete-signal space DCMM, similar to that which has been described in Chapter 6 this dissertation, is proposed in [22] and [23].

[8]For example the probability of a downgrade following a downgrade is likely to be higher than an upgrade following a downgrade, and vice versa.

[9]A BWA approach is used in [22] to estimate model parameters.

observations). The fit of these models was assessed according to the AIC and BIC measures. The analysis performed in [22] showed the most significant model was a DCMM with 3 hidden states in a second order dependence structure, and first order dependence structure in the Markov chain describing the observations. It can be seen in [22] that the credit rating transition matrix for each hidden state ($\hat{\mathbf{B}}^{(1)}$, $\hat{\mathbf{B}}^{(2)}$ and $\hat{\mathbf{B}}^{(3)}$) are clearly different. This indicates that within the credit cycle, there are indeed different risk situations which influence the probabilities of a rating migration.

The possibility of rating drift in a credit rating transition process was mentioned earlier. Analysis performed in [22] indeed confirms the presence of rating drift (by comparing the order one Markov chain with higher order Markov chains). However the final selected DCMM had a first order dependence structure in the Markov chain describing the observations (credit ratings). To explain this, it is proposed by [22] that a credit rating transition process is influenced more significantly by two successive risk situations (hidden states) than by two successive rating observations - hence the second order in the hidden states of the DCMM.

To end off the discussion of [22], using a DCMM approach has provided valuable insights into the varying dynamics of credit rating migrations over time and has also catered for rating drift, non-stationary behaviour present in a credit ratings process, and the influence of the hidden risk states of the credit cycle on the credit ratings. However, the model does have shortfalls (as noted in [22]). In particular, due to the fact that higher orders are used in the final DCMM chosen, the number of parameters in the model is high (152 parameters). It is also noted in [22] that the prediction of future credit ratings becomes challenging if no information of the future risk situations (states) is available (estimating the probabilities that future signals will be observed for a DCMM was discussed in Section 6.2 of this dissertation). Within the credit risk framework, typically expected and unexpected loss needs to be forecast for the next 12 months, hence forecasted future credit ratings are desired. This area of further study noted in [22] is addressed in [23].

As was mentioned earlier, the use of the DCMM in [23] is aligned to [22]. However, in [23] the area of focus is using a Bayesian approach to estimate the DCMM model parameters and the hidden state process. As analysis of rating drift is not of primary focus in [23], a first order (in both the state and signal processes) DCMM is considered.

Similar to [22], the data used in [23] consisted of rated companies over several years of rating history. In the case of [23] the rated companies were restricted to financial institutions and insurance companies (3,918 firms) over a rating history of January 1981 to January 2010. Two hidden states are assumed for the DCMM, and upon reviewing the estimated rating transition probability matrices for the two hidden states (that is $\hat{\mathbf{B}}^{(1)}$ and $\hat{\mathbf{B}}^{(2)}$), it is clear that the first hidden state corresponds to a 'contraction' regime while the second state corresponds to an 'expansion' regime. This is also made clear by comparing these transition matrices to the transition probability matrix of a simple Markov chain model fit to the data.

Furthermore, the estimated DCMM model parameters were used to estimate the hidden state process. As desired, the state process was typically estimated to be in the first state over time periods corresponding to known economic downturns in the financial services industry.

One of the areas of further research identified in [22] was the prediction of future credit ratings using the DCMM estimated rating transition probability matrices. This is explored in [23]. By making use of $\hat{\mathbf{B}}^{(1)}$ and $\hat{\mathbf{B}}^{(2)}$, the expected proportion of defaults (also known as default rates) for each credit rating can be predicted 12 months into the future. As was mentioned when [22] was discussed, this can become quite challenging if no information of the hidden states is available for future time points. This is bypassed in [23] by making different assumptions for the hidden state path over the next 12 months and predicting the default rates under these different scenarios. That is a range of default rates is predicted. For example, one scenario is companies migrate according to the estimated DCMM, conditional on all of the next 12 months

migrating under state 1 (the worst possible scenario); another scenario is companies migrate according to the estimated DCMM, conditional on all of the next 12 months migrating under state 2 (the best possible scenario).

What may also be of interest from a practitioner's point of view is an 'out-of-time' type analysis. In the context of [23], one such test might be to estimate the DCMM model parameters using data up to January 2009. This then leaves a 12 month period to compare the rating migrations and default rates predicted by the DCMM to what actually occurred over the 12 month period (February 2009 to January 2010).

Finally areas of future research are also identified in [23]. These include imposing restrictions when estimating the DCMM parameters (e.g. a practitioner might want to force a credit rating to be absorbing), and to enhance the estimation algorithms to cater for missing data (i.e. if there are gaps in some of the observation sequences). Finally, as was mentioned when discussing [22], as each rated company will give rise to an observation sequence, multiple observation sequences are available to train the DCMM and obtain estimated parameters. These multiple observation sequences are assumed conditionally independent given the hidden state process. It is proposed in [23] that a more complicated data structure between these multiple observation sequences can be explored (e.g. different types of companies might be affected differently during the different periods of the economic/credit cycle).

This then concludes Chapter 8, in which an overview of some of the real-life applications of the HMM and the DCMM (with focus on credit risk) was provided.

Chapter 9

Concluding Remarks

This dissertation has examined two extensions to classical Markov models, namely the hidden Markov model (HMM) and the double chain Markov model (DCMM). These models assume an underlying state process which possesses the Markov property, but which is at no point visible. Instead the output of another process is observed, the distribution of which depends on the state of the model at the time point in question. Hence HMMs and DCMMs assume an observed process which is dependent on an underlying latent Markov process.

A detailed review of these two models has been provided in this dissertation. While different specifications of HMMs and DCMMs have been discussed, the research presented has primarily focused on summarising the literature of the discrete-time, discrete-state space and discrete-signal space HMM and DCMM. Central themes of this dissertation have been establishing the mathematical framework for these models, discussing statistical properties, discussing estimation techniques for the unknown parameters and the hidden state process, and discussing considerations which practitioners of these models would typically need to take into account. In addition, mathematical derivations of key HMM and DCMM results are provided in the appendices of this dissertation. Several of these derivations were, at the time of writing, not found elsewhere in the literature.

Simulation exercises using a two-state two-signal HMM and a two-state two-signal DCMM were presented. These simulations provided useful insights into the mechanics, the effectiveness and the shortcomings of the BWA and the VA to respectively estimate the model parameters and the underlying hidden state sequence. Included in these simulation exercises were studies examining (i) the influence of the starting values on the final BWA estimates, (ii) the influence of the length of the signal sequence on the final BWA estimates, (iii) the accuracy of the BWA in recovering the actual model parameters, (iv) sampling distributions of the BWA parameter estimates, (v) the accuracy of the VA in recovering the underlying hidden state sequence when either actual parameter values or BWA parameter estimates are used to perform the VA and (vi) how the effectiveness of the BWA and the VA compares between the HMM and the DCMM. Conclusions from these studies were made and are presented at the end of each section within Chapter 7.

In order to appropriately conclude the discussion, selected HMM and DCMM applications were reviewed and assessed in light of the conclusions drawn from the simulation study. Attention was given to the application of HMMs and DCMMs in the field of Credit Risk.

Appendix A

Special Relations Between the Independent Mixture Model, Markov Chain, Hidden Markov Model and Double Chain Markov Model

Previous chapters of this dissertation have mentioned special relations which exist between the time homogeneous, discrete-time and discrete-state Markov chain; the time homogeneous, discrete-time and discrete-state independent mixture model; the time homogeneous, discrete-time, discrete-state and discrete-signal HMM; and the time homogeneous, discrete-time, discrete-state and discrete-signal DCMM. This appendix will prove these relations. The mathematics presented were not found in the literature and were derived specifically for the purposes of this dissertation.

Proving the Markov chain a special case of the HMM

In Section 2.1 it was stated that the Markov chain is a special case of the HMM. This is proven below.

To begin, consider an arbitrary Markov chain $\{X_k : k = 1, 2, \ldots\}$ with state space $S = \{1, 2, \ldots, m\}$, transition probabilities $\mathbf{P} = \{p_{ij}\}$ and initial probabilities $\mathbf{a} = [p_1, p_2, \ldots, p_m]$.

Now consider the HMM where the hidden state process is the above described Markov chain $\{X_k : k = 1, 2, \ldots\}$, the observed signal process is $\{S_k : k = 1, 2, \ldots\}$, the signal space is defined as $\delta = S = \{1, 2, \ldots, m\}$ - that is $\nu_i = i$ for each $i \in \{1, 2, \ldots, m\}$, and the signal probability matrix is defined as the $m \times m$ identity matrix - that is $\mathbf{B} = \mathbf{I}_m$.

It is thus required to show that $\{S_k : k = 1, 2, \ldots\}$, the output process of the constructed HMM, is equivalent to $\{X_k : k = 1, 2, \ldots\}$, the output process of the Markov chain. This is achieved in the proof below by showing that (i) the output process of the constructed HMM possesses the Markov property, (ii) the transition probabilities for the output process of the constructed HMM are equivalent to the transition probabilities for the Markov chain, and (iii) $P(S_k = j) = P(X_k = j)$ for each $j \in \{1, 2, \ldots, m\}$ and $k \in \{1, 2, \ldots\}$.

Recall that for a HMM, where signals have been observed for the first n time points, that S_{n+h} will depend on the state visited at time n. That is (see equation (2.4) for more details)

$$P[S_{n+h} = v_k | S_1, X_1, \ldots, S_n, X_n = i] = P[S_{n+h} = v_k | X_n = i] .$$

As the signals $S_1, \ldots, S_{n-1}$ hold valuable information pertaining to the state process visited by the HMM, and therefore also the state visited at time n, the following will typically not hold true for a HMM:

$$P(S_{n+h} = j | S_n = s_n, \ldots, S_1 = s_1, \lambda) = P(S_{n+h} = j | S_n = s_n, \lambda) . \tag{A.1}$$

However in the constructed HMM which is being considered, for $j \in \{1, 2, \ldots, m\}$, $P(S_n = j | X_n = j, \lambda) = 1 = P(X_n = j | S_n = j, \lambda)$. And so the signal observed at n will exactly imply the state visited at time n. Hence equation (A.1) will hold for the

constructed HMM and thus the output process of the constructed HMM possess the Markov property.

The forward equations for the above HMM can be expressed, for each $j \in \{1, 2, \ldots, m\}$ as follows:

$$F_1(j) = p_j\, b_{j,s_1} = \begin{cases} p_{s_1}\,, & \text{if } j = s_1 \\ 0\,, & \text{otherwise} \end{cases}$$

$$F_2(j) = b_{j,s_2} \sum_{i \in S} F_1(i)\, p_{ij} = b_{j,s_2}\, [F_1(s_1)\, p_{s_1,j} + 0] = \begin{cases} p_{s_1}\, p_{s_1,s_2}\,, & \text{if } j = s_2 \\ 0\,, & \text{otherwise} \end{cases}$$

$$F_3(j) = b_{j,s_3} \sum_{i \in S} F_2(i)\, p_{ij} = b_{j,s_3}\, [F_2(s_2)\, p_{s_2,j} + 0] = \begin{cases} p_{s_1}\, p_{s_1,s_2}\, p_{s_2,s_3}\,, & \text{if } j = s_3 \\ 0\,, & \text{otherwise} \end{cases}$$

$$\vdots$$

$$F_n(j) = \begin{cases} p_{s_1}\, p_{s_1,s_2}\, p_{s_2,s_3} \cdots p_{s_{n-1},s_n}\,, & \text{if } j = s_n \\ 0\,, & \text{otherwise.} \end{cases}$$

Now by equation (3.19),

$$\begin{aligned} P(X_{n+h} = j | \mathbf{S}_n = \mathbf{s}_n) &= \frac{1}{\sum_{l \in S} F_n(l)} \sum_{i \in S} p_{ij}(h)\, F_n(i) \\ &= \frac{1}{F_n(s_n)}\, [p_{s_n,j}(h)\, F_n(s_n)] \\ &= p_{s_n,j}(h)\,. \end{aligned}$$

Using this result and equations (3.20) and (A.1), the following can be obtained

$$\begin{aligned} P(S_{n+h} = j | S_n = s_n) &= P(S_{n+h} = j | S_n = s_n, \ldots, S_1 = s_1, \lambda) \\ &= \sum_{i \in S} b_{ij}\, P(X_{n+h} = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\ &= \sum_{i \in S} b_{ij}\, p_{s_n,i}(h) \\ &= 1\,.\, p_{s_n,j}(h) + 0 \\ &= p_{s_n,j}(h) \\ &= P(X_{n+h} = j | X_n = s_n)\,. \end{aligned}$$

And so the transition probabilities for the output process of the constructed HMM are equivalent to the transition probabilities for the Markov chain.

Finally, it follows by the construction of the HMM that $P(S_k = j) = P(X_k = j)$ for each $j \in \{1, 2, \ldots, m\}$ and $k \in \{1, 2, \ldots\}$.

And so, from the above, the output process of the constructed HMM is equivalent to the Markov chain, thereby proving that the Markov chain is indeed a special case of the HMM.

Proving the Independent Mixture Model a special case of the HMM

The independent mixture model was described in Section 2.2. In this section it was stated that the independent mixture model is a special case of the HMM. This is proven below.

To begin, assume an independent mixture model with m states and p_i the probability of selecting state i (where $i = 1, 2, \ldots, m$) for a given time point. By definition $\sum_{i=1}^{m} p_i = 1$. Also assume that given the model is in state i, the observed signal will be emitted according to some distribution f_i.

Also assume a distribution HMM with state process $\{X_k : k = 1, 2, \ldots\}$; state space $S = \{1, 2, \ldots, m\}$; initial state probabilities $\mathbf{a} = [P(X_1 = 1), \ldots, P(X_1 = m)] = [p_1, \ldots, p_m]$; and transition probability matrix

$$\mathbf{P} = \begin{pmatrix} p_1 & p_2 & \cdots & p_m \\ p_1 & p_2 & \cdots & p_m \\ \vdots & \vdots & \ddots & \vdots \\ p_1 & p_2 & \cdots & p_m \end{pmatrix}.$$

Given that the HMM is in state i, assume that the observed signal will be emitted according to the distribution f_i.

Now, in order to show that the above constructed HMM is equivalent to the inde-

pendent mixture model, it must be shown that (for a given time point $k \in \{1, 2, \ldots\}$, states $i, j \in S$ and positive integer h) the following will hold true for the constructed HMM:

$$P(X_k = i) = p_i \tag{A.2}$$

$$P(X_{k+h} = i \,|\, X_k = j) = p_i \,. \tag{A.3}$$

Notice that $\mathbf{P}^2 = \mathbf{P} \,.\, \mathbf{P} = \mathbf{P}$ (since $\sum_{i=1}^{m} p_i = 1$). And so by induction, for some positive integer h, $\mathbf{P}^h = \mathbf{P}$. As was proven in equation (1.6), the h-step transition probability matrix can be obtained by multiplying $\mathbf{P}$ by itself h times. And so $\{p_{ji}(h)\} = \mathbf{P}^{(h)} = \mathbf{P}^h = \mathbf{P} = \{p_i\}$. Equation (A.3) is thus satisfied for the constructed HMM.

Also notice that $\mathbf{a} \,.\, \mathbf{P}^{k-1} = \mathbf{a} \,.\, \mathbf{P} = \mathbf{a}$ (since $\sum_{i=1}^{m} p_i = 1$). And so (by equation (1.8))

$$[P(X_k = 1), \ldots, P(X_k = m)] = \mathbf{p}(k) = \mathbf{a} \,.\, \mathbf{P}^{k-1} = \mathbf{a} = [p_1, \ldots, p_m] \,.$$

And so equation (A.2) is satisfied for the constructed HMM. It is thus proven that the independent mixture model is a special case of the HMM.

Proving the Markov chain a special case of the DCMM

In Section 6.1 it was stated that the Markov chain is a special case of the DCMM. This is proven below.

To begin, consider an arbitrary Markov chain $\{X_k : k = 0, 1, 2, \ldots\}$ with state space $S = \{1, 2, \ldots, m\}$, transition probabilities $\mathbf{P} = \{p_{ij}\}$ and initial probabilities $\mathbf{a} = [p_1, p_2, \ldots, p_m]$.

Now consider the DCMM with hidden state process $\{Y_k : k = 1, 2, \ldots\}$ where the state space is defined as $\tilde{S} = \{1\}$. That is $P(Y_k = 1) = 1$ for each $k = 1, 2, \ldots$ Further suppose that the observed signal process is $\{S_k : k = 0, 1, 2, \ldots\}$, the signal

space is defined as $\delta = S = \{1, 2, \ldots, m\}$ - that is $\nu_i = i$ for each $i \in \{1, 2, \ldots, m\}$, and $\mathbf{B}^{(1)} = \mathbf{P}$.

It is thus required to show that $\{S_k\}$, the output process of the DCMM, is equivalent to $\{X_k\}$, the output process of the Markov chain. This is achieved in the proof below by showing that (i) the output process of the constructed DCMM possesses the Markov property, (ii) the transition probabilities for the output process of the constructed DCMM are equivalent to the transition probabilities for the Markov chain, and (iii) $P(S_k = i) = P(X_k = i)$ for each $i \in \{1, 2, \ldots, m\}$ and $k \in \{0, 1, 2, \ldots\}$.

Firstly note that by definition of the DCMM, $\{S_k\}$ possesses the Markov property.

Further note that by construction of the DCMM in question, the following will hold for each $i, j \in \{1, 2, \ldots, m\}$:

$$P(S_{k+1} = j | S_k = i) = P(S_{k+1} = j | S_k = i, Y_k = 1) = b_{ij}^{(1)} = p_{ij} \, .$$

And so the transition probabilities for the output process of the constructed DCMM are equivalent to the transition probabilities for the Markov chain.

In Section 6.1 it was explained that an initial signal at time 0 is considered for the DCMM with no corresponding hidden state. Assume that the initial signal for the DCMM is chosen so that it possesses the distribution defined by $\mathbf{a}$. That is $P(S_0 = i) = p_i = P(X_0 = i)$ for each $i \in \{1, 2, \ldots, m\}$.

And so both the initial probabilities and transition probabilities are equivalent for $\{X_k\}$ and $\{S_k\}$. It can then be easily verified from equation (1.8) that $P(X_k = i) = P(S_k = i)$ for $k = 0, 1, 2$.

And so, from the above, the output process of the constructed DCMM is equivalent to the output process of the Markov chain. This then proves that the Markov chain is indeed a special case of the DCMM.

Proving the HMM a special case of the DCMM

In Section 6.1 it was stated that the HMM is a special case of the DCMM. This is proven below.

To begin consider a HMM with state process $\{X_k\}$, state space $S = \{1, 2, \ldots, m\}$, signal process $\{Y_k\}$, signal space $\delta = \{\nu_1, \nu_2, \ldots, \nu_M\}$, and signal probability matrix $\mathbf{B}$ given as follows

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mM} \end{pmatrix}.$$

Now consider a DCMM with state process $\{X_k\}$, signal process $\{S_k\}$ and signal space δ. For state $i \in S$, assume that the signal transition probability matrix is given by

$$\mathbf{B}^{(i)} = \begin{pmatrix} b_{i1} & b_{i2} & \cdots & b_{iM} \\ b_{i1} & b_{i2} & \cdots & b_{iM} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{iM} \end{pmatrix}.$$

It is thus required to show that $\{S_k\}$, the output process of the constructed DCMM, is equivalent to $\{Y_k\}$, the output process of the HMM. This is achieved in the proof below by showing that $P(S_k = \nu_j) = P(Y_k = \nu_j)$ for $\nu_j \in \delta$ and $k \in \{1, 2, \ldots\}$.

Notice from the above that for $k \in \{1, 2, \ldots\}$, $i \in S$ and $\nu_j, \nu_l \in \delta$ the following holds

$$P(S_k = \nu_j \,|\, X_k = i, S_{k-1} = \nu_l) = b_{lj}^{(i)} = b_{ij}\,.$$

Further notice that

$$\begin{aligned}
P(S_k = \nu_j \,|\, X_k = i) &= \sum_{\nu_l \in \delta} P(S_k = \nu_j \,, S_{k-1} = \nu_l \,|\, X_k = i) \quad \text{.....} \quad \text{by (2.9)} \\
&= \sum_{\nu_l \in \delta} P(S_{k-1} = \nu_l \,|\, X_k = i)\, P(S_k = \nu_j \,|\, X_k = i \,, S_{k-1} = \nu_l) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{.....} \qquad \text{by (2.11)} \\
&= \sum_{\nu_l \in \delta} P(S_{k-1} = \nu_l \,|\, X_k = i)\, b_{ij} \\
&= b_{ij} \sum_{\nu_l \in \delta} P(S_{k-1} = \nu_l \,|\, X_k = i) \\
&= b_{ij} \,.\, 1 \\
&= b_{ij} \\
&= P(Y_k = \nu_j \,|\, X_k = i) \,.
\end{aligned}$$

And so

$$\begin{aligned}
P(S_k = \nu_j) &= \sum_{i \in S} P(S_k = \nu_j \,|\, X_k = i) P(X_k = i) \quad \text{.....} \quad \text{by (2.9) and (2.10)} \\
&= \sum_{i \in S} P(Y_k = \nu_j \,|\, X_k = i) P(X_k = i) \\
&= P(Y_k = \nu_j) \quad \text{.....} \quad \text{by (2.9) and (2.10).}
\end{aligned}$$

From the above it can be seen that $\{S_k\}$, the output process of the constructed DCMM, is equivalent to $\{Y_k\}$, the output process of the HMM. This then confirms that the HMM is indeed a special case of the DCMM.

Appendix B

Further Discussions Surrounding the Baum-Welch Algorithm

The Baum-Welch algorithm (BWA) for the HMM was discussed in Section 4.1 of this dissertation. In particular, in Section 4.1.1, the Baum-Welch re-estimation equations were derived by interpreting the summations of certain probabilities as the expected number of occurrences of events. These interpretations are formally proven in Section B.1 of this appendix. Also mentioned in Section 4.1.1 was that the Baum-Welch re-estimation equations can alternatively be derived through the use of the Expected Maximization (EM) algorithm. The details substantiating this are given in Section B.2 of this appendix.

B.1 Proof of Results Used in the Baum-Welch Algorithm for the HMM

Equation (4.5) of Section 4.1.1 stated how $\sum_{k=1}^{n} \gamma_k(i)$, $\sum_{k=1}^{n-1} \gamma_k(i)$, $\sum_{k=1}^{n-1} \xi_k(i,j)$ and $\sum_{k=1}^{n} \gamma_{k,m}(i)$ can be interpreted as the expected number of occurrences of certain events for the first n time points. This section will formally prove these results. It should be noted that these proofs were not found in any references within the literature but were de-

rived specifically for this dissertation.

To begin, recall that $\mathbf{S}_n = (S_1, \ldots, S_n)$ is the vector of random variables for the first n signals, and $\mathbf{s}_n = (s_1, \ldots, s_n)$ is the actual sequence of the first n signals which have been observed, where $s_k \in \delta$ for $k = 1, 2 \ldots, n$. Now,

$$
\begin{aligned}
& P(\text{Process is in state } i \text{ at time } k \text{ and the } k^{th} \text{ signal is } \nu_m | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
= \;& P(X_k = i, S_k = \nu_m | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
= \;& P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda)\ P(S_k = \nu_m | X_k = i, \mathbf{S}_n = \mathbf{s}_n, \lambda) \quad \ldots\ldots \quad \text{by (2.11)} \\
= \;& P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) \times \begin{cases} 1 & \text{if} \quad s_k = \nu_m \\ 0 & \text{if} \quad s_k \neq \nu_m \end{cases} \\
= \;& \begin{cases} P(X_k = i | \mathbf{S}_n = \mathbf{s}_n, \lambda) & \text{if} \quad s_k = \nu_m \\ 0 & \text{if} \quad s_k \neq \nu_m \end{cases} \\
= \;& \begin{cases} \gamma_k(i) & \text{if} \quad s_k = \nu_m \\ 0 & \text{if} \quad s_k \neq \nu_m \end{cases} \quad \ldots\ldots \quad \text{by (4.3)} \\
= \;& \gamma_{k,m}(i)\,.
\end{aligned}
\tag{B.1}
$$

Also, recall the following well known statistical result (see for example page 150 of [17])

$$
\mathrm{E}(g(Z) | Y = y) = \sum_z g(z)\,.\,P(Z = z | Y = y)\,. \tag{B.2}
$$

Let X be a random variable representing the number of times the HMM is in state i during the first n time points, and let

$$
X_{i,k} = \begin{cases} 1, & \text{if the HMM is in state } i \text{ at time } k \\ 0, & \text{otherwise.} \end{cases}
$$

So, $X = \sum_{k=1}^{n} X_{i,k}\,.$

Now,

Expected number of times the HMM is in state i during the first n observed time points

$$
\begin{aligned}
&= \mathrm{E}(X|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \mathrm{E}(\sum_{k=1}^{n} X_{i,k}|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} \mathrm{E}(X_{i,k}|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} [(1)P(X_{i,k} = 1|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \,+\, (0)P(X_{i,k} = 0|\,\mathbf{S}_n = \mathbf{s}_n, \lambda)] \quad ..... \quad \text{by } (B.2) \\
&= \sum_{k=1}^{n} P(X_{i,k} = 1|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} P(\text{The HMM is in state } i \text{ at time } k|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} P(X_k = i|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} \gamma_k(i) \quad ..... \quad \text{by (4.3).}
\end{aligned}
$$

And so the first result given in equation (4.5) is proven.

Now define Y to be the number of transitions by the HMM from state i during the first n time points, and

$$Y_{i,j,k} = \begin{cases} 1, & \text{if the HMM is in state } i \text{ at time } k \text{ and state } j \text{ at time } k+1 \\ 0, & \text{otherwise.} \end{cases}$$

Then $Y = \sum_{k=1}^{n-1} \sum_{j \in S} Y_{i,j,k}$ is satisfied.

Using similar mathematics to above, the following is obtained:

Expected number of transitions from state i during the first n observed time points

$$
\begin{aligned}
&= \mathrm{E}(Y|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n-1} \sum_{j \in S} [(1)P(Y_{i,j,k} = 1|\mathbf{S}_n = \mathbf{s}_n, \lambda) \; + \; (0)P(Y_{i,j,k} = 0|\mathbf{S}_n = \mathbf{s}_n, \lambda)] \\
&= \sum_{k=1}^{n-1} \sum_{j \in S} P(X_k = i, X_{k+1} = j|\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n-1} P(X_k = i|\mathbf{S}_n = \mathbf{s}_n, \lambda) \quad ..... \quad \text{by (2.9)} \\
&= \sum_{k=1}^{n-1} \gamma_k(i) \quad ..... \quad \text{by (4.3).}
\end{aligned}
$$

And so the second result given in equation (4.5) is proven.

By defining Z to be the number of transitions by the HMM from state i to state j during the first n time points, $Z = \sum_{k=1}^{n-1} Y_{i,j,k}$ is satisfied.

Using similar mathematics to above, the following is obtained:

Expected number of transitions from state i to j during the first n observed time points

$$
\begin{aligned}
&= \mathrm{E}(Z|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n-1} P(X_k = i, X_{k+1} = j|\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n-1} \xi_k(i,j) \quad ..... \quad \text{by (4.3).}
\end{aligned}
$$

And so the third result given in equation (4.5) is proven.

Finally, define W to be the number of times the HMM is in state i and emits signal ν_m during the first n time points, and

$$
W_{i,m,k} = \begin{cases} 1, & \text{if the HMM is in state } i \text{ and emits signal } \nu_m \text{ at time } k \\ 0, & \text{otherwise.} \end{cases}
$$

Then $W = \sum_{k=1}^{n} W_{i,m,k}$ is satisfied.

Using similar mathematics to above, the following is obtained:

Expected number of times the HMM is in state i and emits signal ν_m during the first n observed time points

$$
\begin{aligned}
&= \mathrm{E}(W|\,\mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} P(X_k = i, S_k = \nu_m | \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} \gamma_{k,m}(i) \quad \text{.....} \quad \text{by } (B.1).
\end{aligned}
$$

And so the final result given in equation (4.5) is proven.

B.2 Relation of the Baum-Welch Algorithm to the EM framework

Application of the EM algorithm to the estimation of the HMM model parameters yields identical re-estimation equations to that of the BWA. This is detailed in Section B.2.2. As an introduction to this, a general discussion on the EM algorithm is first be presented in Section B.2.1.

B.2.1 The EM Algorithm

The EM algorithm was formally presented and named for the first time in [20] - a paper published by Dempster, Laird and Rubin in 1977. It was however noted in [20] that the method had been "proposed many times in special circumstances" by earlier authors. For example, a detailed treatment of the EM method for exponential families had been published in several papers authored by R. Sundberg. However, [20] generalized the methodology and detailed a convergence analysis for a wider

class of problems. The algorithm presented in [20] has since been expanded upon by subsequent research, particularly regarding convergence analysis of the algorithm. Since these initial papers, the EM algorithm has been established as an important tool in statistical analysis, and the algorithm has been described and made use of in numerous publications. Two such publications are [13] and [17], from which the work presented in this section is predominantly adapted.

In short, the EM algorithm is an iterative procedure for finding the maximum likelihood estimators (MLEs) of parameters in statistical models which depend on unobserved or latent variables.

To begin, let $\mathbf{X}$ be an independent and identically distributed (iid) random sample which has been observed. The MLE of the parameter set θ is then the value of θ which will maximise the likelihood function $P(\mathbf{X}|\,\theta)$. So, the aim of maximum likelihood estimation is to estimate the model parameter(s) for which the observed data is most likely. In order to simplify the mathematics when finding the MLE, it is typical to introduce the log likelihood function defined as

$$L(\theta) = \ln P(\mathbf{X}|\,\theta)\,.$$

Since $ln(x)$ is a strictly increasing function, the value of θ which maximises $L(\theta)$ will also maximise $P(\mathbf{X}|\,\theta)$.

Now, the EM algorithm is an iterative procedure for maximising $L(\theta)$ when the random sample contains both observed and unobserved or latent variables. Denote the random vector for the unobserved variables by $\mathbf{Z}$ and a given realisation by $\mathbf{z}$. Using equations (2.9) and (2.11), the total likelihood function can be written to incorporate the hidden variables $\mathbf{z}$ as follows

$$P(\mathbf{X}|\,\theta) = \sum_{\mathbf{z}} P(\mathbf{X}|\,\mathbf{z},\theta)\,P(\mathbf{z}|\,\theta)\,.$$

Since the EM algorithm is an iterative procedure, assume that after the n^{th} iteration the current estimate for θ is given by θ_n. Since the objective is to maximise $P(\mathbf{X}|\,\theta)$, it

is essential that the updated estimate calculated by the EM algorithm, θ_{n+1} , satisfies the following

$$L(\theta_{n+1}) \geq L(\theta_n)\,.$$

Now, consider the function $l(\theta|\,\theta_n)$ which is defined as follows

$$l(\theta|\,\theta_n) = L(\theta_n) + \sum_{\mathbf{z}} P(\mathbf{z}|\,\mathbf{X},\theta_n) \ln\left(\frac{P(\mathbf{X}|\,\mathbf{z},\theta)\,P(\mathbf{z}|\,\theta)}{P(\mathbf{z}|\,\mathbf{X},\theta_n)\,P(\mathbf{X}|\,\theta_n)}\right)\,.$$

It is shown in [13] that this function has the following properties:

$L(\theta) \geqslant l(\theta|\,\theta_n)$ for all θ, and

$L(\theta_n) = l(\theta_n|\,\theta_n)$, that is for $\theta = \theta_n$ the functions $L(\theta)$ and $l(\theta|\,\theta_n)$ are equal.

Recall that the objective of the EM algorithm is to find the value of θ which will maximise $L(\theta)$. This is achieved as follows.

Consider a value for θ, denoted $\tilde{\theta}$, which satisfies

$$l(\tilde{\theta}|\,\theta_n) \geq l(\theta_n|\,\theta_n)\,.$$

Then, using the properties stated above, the following is obtained

$$L(\tilde{\theta}) \geq l(\tilde{\theta}|\,\theta_n) \geq l(\theta_n|\,\theta_n) = L(\theta_n)\,.$$

Therefore, if θ_{n+1} is chosen such that $l(\theta_{n+1}|\,\theta_n) \geq l(\theta_n|\,\theta_n)$, it can be guaranteed that $L(\theta_{n+1}) \geq L(\theta_n)$ will hold true for each step of the iterative procedure. So, for each iteration, the log likelihood is non-decreasing - thereby ensuring the desired property for finding the value of θ which will maximise $L(\theta)$.

In order to achieve the greatest possible increase in the value of $L(\theta)$ at each iteration, the EM algorithm will select θ_{n+1} to be the value of θ which will maximise $l(\theta|\,\theta_n)$. That is, θ_{n+1} is selected by the EM algorithm as follows:

$$\theta_{n+1} = \arg\max_{\theta}\left\{l(\theta|\,\theta_n)\right\}\,.$$

This single iteration of the EM algorithm is illustrated in Figure B.1 (reproduced from [13]).

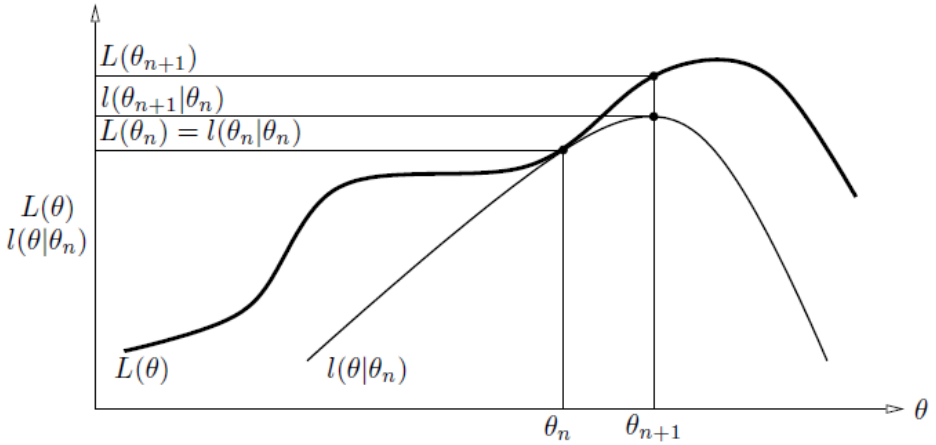


Figure B.1: Graphical interpretation of a single iteration of the EM algorithm: The function $l(\theta|\,\theta_n)$ is bounded above by the likelihood function $L(\theta)$. The functions are equal at $\theta = \theta_n$. The EM algorithm chooses θ_{n+1} as the value of θ for which $l(\theta|\,\theta_n)$ is a maximum. This ensures that the value of the likelihood function $L(\theta)$ is increased at each step.

So, the EM algorithm produces the following,

$$\begin{aligned}
\theta_{n+1} = \quad & \arg\max_{\theta} \{l(\theta|\,\theta_n)\} \\
= \quad & \arg\max_{\theta} \left\{ L(\theta_n) + \sum_{\mathbf{z}} P(\mathbf{z}|\,\mathbf{X}, \theta_n) \ln \left(\frac{P(\mathbf{X}|\,\mathbf{z}, \theta)\, P(\mathbf{z}|\,\theta)}{P(\mathbf{z}|\,\mathbf{X}, \theta_n)\, P(\mathbf{X}|\,\theta_n)} \right) \right\} \\
= \quad & \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} P(\mathbf{z}|\,\mathbf{X}, \theta_n) \ln \left(P(\mathbf{X}|\,\mathbf{z}, \theta)\, P(\mathbf{z}|\,\theta) \right) \right\} \\
= \quad & \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} P(\mathbf{z}|\,\mathbf{X}, \theta_n) \ln P(\mathbf{X}, \mathbf{z}|\,\theta) \right\} \quad \text{.....} \quad \text{by (2.11)} \qquad \text{(B.3)} \\
= \quad & \arg\max_{\theta} \left\{ \mathrm{E}_{\mathbf{Z}|\mathbf{X}, \theta_n} \left[\ln P(\mathbf{X}, \mathbf{Z}|\,\theta) \right] \right\} \quad \text{.....} \quad \text{by } (B.2). \qquad \text{(B.4)}
\end{aligned}$$

Note that in some literature $\mathrm{E}_{\mathbf{Z}|\mathbf{X}, \theta_n} \left[\ln P(\mathbf{X}, \mathbf{Z}|\,\theta) \right]$ is equivalently notated as

$$Q(\theta, \theta_n) = \mathrm{E}_{\mathbf{Z}} [\ln P(\mathbf{X}, \mathbf{Z}|\,\theta)\,|\,\mathbf{X}, \theta_n]\,.$$

From equation (B.4) the iterating E (Expectation) and M (Maximization) steps of the EM algorithm become clear:

E-step: Calculate $\mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta_n}[\ln P(\mathbf{X},\mathbf{z}|\,\theta)]$ - the expected value of the log likelihood function with respect to the conditional distribution of $\mathbf{Z}$ (the unknown hidden data) given $\mathbf{X}$ (the observed sample data) and θ_n (the current estimate of θ). That is, $\mathbf{Z}$ is a random variable governed by the distribution $P(\mathbf{z}|\,\mathbf{X},\theta_n)$, where $\mathbf{X}$ and θ_n are viewed as constants.

In some applications of the EM algorithm, the above expectation may be difficult to calculate. In such instances it may be computationally simpler to use the equivalent expression given in equation (B.3). This is often true when the EM algorithm is used in the context of the HMM, as will be shown in the next section.

M-step: Maximise either expression (B.3) or (B.4) with respect to θ.

At this point it is fair to ask what has been gained in the MLE calculation given that we have simply traded the maximization of $L(\theta)$ for the maximization of $l(\theta|\,\theta_n)$. The answer lies in the fact that $l(\theta|\,\theta_n)$ takes into account the unobserved or hidden data $\mathbf{Z}$. In the case where it is desired to take $\mathbf{Z}$ into account when calculating the MLE, the EM algorithm provides a framework for doing so.

Details of the convergence properties of the EM algorithm can be viewed in [36] and are also summarised in [13] and [17]. In [13] it is stated that the EM algorithm will converge to a stationary point of the likelihood function but that this stationary point is not guaranteed be a local maximum. To this end [13] notes that "it is possible for the algorithm to converge to local minima or saddle points in unusual cases". However, [17] indicates that under appropriate conditions (see page 370) convergence to a local maximum or saddle point is guaranteed.

B.2.2 Using the EM Algorithm to Estimate the Parameters of a HMM

Section 4.1.1 of this dissertation detailed how the parameters of the time homogeneous, discrete-time, discrete-state and discrete-signal HMM can be estimated using the Baum-Welch Algorithm (BWA). This section will detail the mathematics which show that these estimates are identical to the estimates obtained when the EM algorithm is applied to this HMM. In so doing, it will thus be shown that the estimates produced by the BWA are indeed MLEs.

While the work presented in this section is predominately adapted from [12], certain mathematical details from [12] have been expanded upon in this dissertation to provide additional clarity.

To begin, recall from the previous section that the EM algorithm is an iterative procedure which is used to find the MLE of parameters in statistical models which contain unobserved or latent data. Each iteration of the algorithm can be performed by making use of the following:

$$\begin{aligned} \theta_{n+1} &= \arg\max_{\theta} \{Q(\theta, \theta_n)\} \\ &= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} P(\mathbf{z}|\, \mathbf{X}, \theta_n) \ln P(\mathbf{X}, \mathbf{z}|\, \theta) \right\}, \end{aligned} \tag{B.5}$$

where $\mathbf{X}$ represents the observed data and $\mathbf{z}$ represents the unobserved data.

In order to ease the computations which are to follow (while as far as possible keeping the notation consistent to that which has been previously used in this dissertation),

consider the following notation:

λ denote the set of unknown parameters for the HMM which need to be estimated,

λ^* denote the current estimate of λ,

$\hat{\lambda}$ denote the updated estimate of λ,

$\mathbf{X} = (X_1\,, X_2\,, \ldots\,, X_n)$ be the random vector denoting the hidden states visited by the HMM during the first n time points (previously notated as $\mathbf{X}_n$),

$\mathbf{x} = (i_1\,, i_2\,, \ldots\,, i_n)$ be a realisation of $\mathbf{X}$,

$\mathbf{S} = (S_1\,, S_2\,, \ldots\,, S_n)$ be the random vector denoting the signals observed during the first n time points (previously notated as $\mathbf{S}_n$),

$\mathbf{s} = (s_1\,, s_2\,, \ldots\,, s_n)$ be a realisation of $\mathbf{S}$,

α denote the state space (previously notated as S),

δ denote the signal space.

Using this notation, equation (B.5) can be written for the HMM as:

$$\begin{aligned} \hat{\lambda} &= \arg\max_{\lambda} \{Q(\lambda, \lambda^*)\} \\ &= \arg\max_{\lambda} \left\{ \sum_{\mathbf{x}\in\mathbf{X}} P(\mathbf{x}|\,\mathbf{S}, \lambda^*) \ln P(\mathbf{S}, \mathbf{x}|\,\lambda) \right\}, \end{aligned} \tag{B.6}$$

where

$$\sum_{\mathbf{x}\in\mathbf{X}} P(\mathbf{x}) \equiv \sum_{i_1\in\alpha} \sum_{i_2\in\alpha} \cdots \sum_{i_n\in\alpha} P(\mathbf{X} = (i_1, i_2, \ldots, i_n))\,.$$

By making use of equation (3.10), the following can be derived:

$$\ln P(\mathbf{S}, \mathbf{x}|\,\lambda) = \ln(p_{i_1}) + \sum_{t=1}^{n-1} \ln(p_{i_t, i_{t+1}}) + \sum_{t=1}^{n} \ln(b_{i_t, s_t})\,.$$

Using this, the function $Q(\lambda, \lambda^*)$ can then be written as follows:

$$Q(\lambda, \lambda^*) = \sum_{\mathbf{x}\in\mathbf{X}} \ln(p_{i_1}) P(\mathbf{x}|\,\mathbf{S}, \lambda^*) + \sum_{\mathbf{x}\in\mathbf{X}} \sum_{t=1}^{n-1} \ln(p_{i_t,i_{t+1}}) P(\mathbf{x}|\,\mathbf{S}, \lambda^*) + \sum_{\mathbf{x}\in\mathbf{X}} \sum_{t=1}^{n} \ln(b_{i_t,s_t}) P(\mathbf{x}|\,\mathbf{S}, \lambda^*)\,. \tag{B.7}$$

Now, the first term of equation (B.7) can be simplified in the following way:

$$\begin{aligned}
& \sum_{\mathbf{x}\in\mathbf{X}} \ln(p_{i_1}) P(\mathbf{x}|\,\mathbf{S}, \lambda^*) \\
= & \sum_{i_1\in\alpha} \sum_{i_2\in\alpha} \cdots \sum_{i_n\in\alpha} \ln(p_{i_1}) P(X_1 = i_1, X_2 = i_2, \ldots, X_n = i_n|\,\mathbf{S}, \lambda^*) \\
= & \sum_{i_1\in\alpha} \ln(p_{i_1}) \sum_{i_2\in\alpha} \cdots \sum_{i_n\in\alpha} P(X_1 = i_1, X_2 = i_2, \ldots, X_n = i_n|\,\mathbf{S}, \lambda^*) \\
= & \sum_{i_1\in\alpha} \ln(p_{i_1})\, P(X_1 = i_1|\,\mathbf{S}, \lambda^*) \quad \ldots\ldots \quad \text{by (2.9)} \\
= & \sum_{k\in\alpha} \ln(p_k)\, P(X_1 = k|\,\mathbf{S}, \lambda^*)\,.
\end{aligned}$$

The second term of equation (B.7) can be simplified as follows:

$$\begin{aligned}
& \sum_{\mathbf{x}\in\mathbf{X}} \sum_{t=1}^{n-1} \ln(p_{i_t,i_{t+1}}) P(\mathbf{x}|\,\mathbf{S}, \lambda^*) \\
= & \sum_{t=1}^{n-1} \sum_{i_1\in\alpha} \cdots \sum_{i_n\in\alpha} \ln(p_{i_t,i_{t+1}}) P(X_1 = i_1, \ldots, X_n = i_n|\,\mathbf{S}, \lambda^*)
\end{aligned} \tag{B.8}$$

(since the indices of the summations are not dependent on each other, the order of summation may be changed).

Now, assume that the index for t is currently at $t^* \in \{1, 2, \ldots, n-1\}$. Then for $t = t^*$, equation (B.8) becomes

$$\begin{aligned}
& \sum_{i_{t^*} \in \alpha} \sum_{i_{t^*+1} \in \alpha} \ln(p_{i_{t^*}, i_{t^*+1}}) \sum_{i_1 \in \alpha} \cdots \sum_{i_{t^*-1} \in \alpha} \sum_{i_{t^*+2} \in \alpha} \cdots \sum_{i_n \in \alpha} P(X_1 = i_1, \ldots, X_n = i_n | \, \mathbf{S}, \lambda^*) \\
= & \sum_{i_{t^*} \in \alpha} \sum_{i_{t^*+1} \in \alpha} \ln(p_{i_{t^*}, i_{t^*+1}}) \, P(X_{t^*} = i_{t^*}, X_{t^*+1} = i_{t^*+1} | \, \mathbf{S}, \lambda^*) \quad \ldots\ldots \quad \text{by (2.9)} \\
= & \sum_{k \in \alpha} \sum_{l \in \alpha} \ln(p_{k,l}) \, P(X_{t^*} = k, X_{t^*+1} = l | \, \mathbf{S}, \lambda^*) \, .
\end{aligned}$$

And so the second term of equation (B.7) can be written as:

$$\sum_{k \in \alpha} \sum_{l \in \alpha} \ln(p_{k,l}) \sum_{t=1}^{n-1} P(X_t = k, X_{t+1} = l | \, \mathbf{S}, \lambda^*) \, .$$

The final term of (B.7) can be similarly simplified to yield

$$\sum_{\mathbf{x} \in \mathbf{X}} \sum_{t=1}^{n} \ln(b_{i_t, s_t}) P(\mathbf{x} | \, \mathbf{S}, \lambda^*) = \sum_{k \in \alpha} \sum_{t=1}^{n} \ln(b_{k, s_t}) P(X_t = k | \, \mathbf{S}, \lambda^*) \, .$$

And so the function $Q(\lambda, \lambda^*)$ has been simplified to the following:

$$\begin{aligned}
Q(\lambda, \lambda^*) = & \sum_{k \in \alpha} \ln(p_k) \, P(X_1 = k | \, \mathbf{S}, \lambda^*) \\
& + \sum_{k \in \alpha} \sum_{l \in \alpha} \ln(p_{k,l}) \sum_{t=1}^{n-1} P(X_t = k, X_{t+1} = l | \, \mathbf{S}, \lambda^*) \\
& + \sum_{k \in \alpha} \sum_{t=1}^{n} \ln(b_{k, s_t}) P(X_t = k | \, \mathbf{S}, \lambda^*) \, . \qquad \text{(B.9)}
\end{aligned}$$

Now recall from equation (B.6) that

$$\hat{\lambda} = \arg\max_{\lambda} \{Q(\lambda, \lambda^*)\} \, .$$

In particular it is desired the parameters of $\hat{\lambda}$ be found subject to

$$\sum_{i\in\alpha} \hat{p}_i = 1$$

$$\sum_{j\in\alpha} \hat{p}_{ij} = 1, \qquad \text{for } i \in \alpha$$

$$\sum_{v_k\in\delta} \hat{b}_{ik} = 1, \qquad \text{for } i \in \alpha\,.$$

This can be achieved by making use of the Lagrange multiplier and setting the partial derivative to zero.

For $\hat{p}_i$, where $i \in \alpha$, this yields the following:

$$\begin{aligned}
\frac{\partial}{\partial p_i}\left[Q(\lambda,\lambda^*) + \beta(\sum_{k\in\alpha} p_k - 1)\right] &= 0 \\
\frac{\partial}{\partial p_i}\left[\sum_{k\in\alpha} \ln(p_k)\, P(X_1 = k|\,\mathbf{S},\lambda^*)\right] + \frac{\partial}{\partial p_i}\left[\beta(\sum_{k\in\alpha} p_k - 1)\right] &= 0 \\
\frac{1}{p_i}\, P(X_1 = i|\,\mathbf{S},\lambda^*) + \beta + 0 &= 0 \\
\Rightarrow \hat{p}_i &= \frac{-P(X_1 = i|\,\mathbf{S},\lambda^*)}{\beta}\,.
\end{aligned}$$

So, from the above,

$$\begin{aligned}
1 &= \sum_{i\in\alpha} \hat{p}_i = -\frac{1}{\beta}\sum_{i\in\alpha} P(X_1 = i|\,\mathbf{S},\lambda^*) = -\frac{1}{\beta}(1) \\
\Rightarrow \beta &= -1 \\
\Rightarrow \hat{p}_i &= P(X_1 = i|\,\mathbf{S},\lambda^*) \\
&= \gamma_1^*(i) \quad \text{.....} \quad \text{by (4.3) and (4.6).}
\end{aligned}$$

And so the estimate for p_i obtained from the BWA (see equation (4.7)) is the same as the above MLE of p_i obtained using the EM algorithm.

To show that this estimate is indeed a local maxima, the second partial derivative

can be evaluated at the above calculated $\hat{p}_i$. Through the use of the quotient rule for differentiation, this yields:

$$\frac{\partial^2}{\partial {p_i}^2}\left[Q(\lambda,\lambda^*)+\beta(\sum_{k\in\alpha}p_k-1)\right]\Bigg|_{p_i=\hat{p}_i}=-\frac{1}{\gamma_1^*(i)}\,.$$

Since $\gamma_1^*(i)\geq 0$ (this was discussed in the paragraph below equation (4.10) in Section 4.1), $-\frac{1}{\gamma_1^*(i)}<0$ for $\gamma_1^*(i)\neq 0$.

If $\gamma_1^*(i)=0$ then inspection of equation (B.9) reveals that the value for p_i which will maximise $Q(\lambda,\lambda^*)$, subject to $\sum_{i\in\alpha}p_i=1$, is $p_i=0=\gamma_1^*(i)=\hat{p}_i$.

And so $\hat{p}_i=\gamma_1^*(i)$ is indeed a local maxima.

Next the MLE for p_{ij}, where $i,j\in\alpha$, is derived as follows:

$$\begin{aligned}
\frac{\partial}{\partial p_{ij}}\left[Q(\lambda,\lambda^*)+\beta(\sum_{k\in\alpha}p_{ik}-1)\right] &= 0\\
\frac{\partial}{\partial p_{ij}}\left[\sum_{k\in\alpha}\sum_{l\in\alpha}\ln(p_{k,l})\sum_{t=1}^{n-1}P(X_t=k,X_{t+1}=l|\,\mathbf{S},\lambda^*)\right]+\beta &= 0\\
\frac{1}{p_{ij}}\left[\sum_{t=1}^{n-1}P(X_t=i,X_{t+1}=j|\,\mathbf{S},\lambda^*)\right]+\beta &= 0\\
\Rightarrow \hat{p}_{ij} &= \frac{-\sum_{t=1}^{n-1}P(X_t=i,X_{t+1}=j|\,\mathbf{S},\lambda^*)}{\beta}.
\end{aligned}$$

So, from the above,

$$\begin{aligned}
1 = \sum_{j\in\alpha}\hat{p}_{ij} &= -\frac{1}{\beta}\sum_{j\in\alpha}\sum_{t=1}^{n-1} P(X_t = i, X_{t+1} = j|\,\mathbf{S},\lambda^*) \\
&= -\frac{1}{\beta}\sum_{t=1}^{n-1}\sum_{j\in\alpha} P(X_t = i, X_{t+1} = j|\,\mathbf{S},\lambda^*) \\
&= -\frac{1}{\beta}\sum_{t=1}^{n-1} P(X_t = i|\,\mathbf{S},\lambda^*) \quad \text{.....} \quad \text{by (2.9)} \\
\Rightarrow \beta &= -\sum_{t=1}^{n-1} P(X_t = i|\,\mathbf{S},\lambda^*) \\
\Rightarrow \hat{p}_{ij} &= \frac{\sum_{t=1}^{n-1} P(X_t = i, X_{t+1} = j|\,\mathbf{S},\lambda^*)}{\sum_{t=1}^{n-1} P(X_t = i|\,\mathbf{S},\lambda^*)} \\
&= \frac{\sum_{t=1}^{n-1}\xi_t^*(i,j)}{\sum_{t=1}^{n-1}\gamma_t^*(i)} \quad \text{.....} \quad \text{by (4.3) and (4.6).}
\end{aligned}$$

And so the estimate for p_{ij} obtained from the BWA (see equation (4.8)) is the same as the above MLE of p_{ij} obtained using the EM algorithm.

Using similar techniques to that which was used for $\hat{p}_i$, it can be verified that $\hat{p}_{ij}$ is indeed a local maxima.

Next the MLE for b_{jm} will be derived. To achieve this define, for $\nu_m \in \delta$ and $t \in \{1, 2, \ldots, n\}$, $\mathrm{I}_t(\nu_m)$ to be the following indicator function:

$$\mathrm{I}_t(\nu_m) = \begin{cases} 1 & \text{if} \quad s_t = \nu_m \\ 0 & \text{if} \quad s_t \neq \nu_m \end{cases} .$$

Then the MLE for b_{jm}, where $j \in \alpha$ and $\nu_m \in \delta$, is derived as follows:

$$\begin{aligned}
\frac{\partial}{\partial b_{jm}} \left[Q(\lambda, \lambda^*) + \beta(\sum_{\nu_k \in \delta} b_{jk} - 1) \right] &= 0 \\
\frac{\partial}{\partial b_{jm}} \left[\sum_{k \in \alpha} \sum_{t=1}^{n} \ln(b_{k,s_t})\, P(X_t = k | \, \mathbf{S}, \lambda^*) \right] + \beta &= 0 \\
\sum_{t=1}^{n} \left(\frac{1}{b_{jm}} \right) \mathrm{I}_t(\nu_m)\, P(X_t = j | \, \mathbf{S}, \lambda^*) + \beta &= 0 \\
\Rightarrow \hat{b}_{jm} &= \frac{-\sum_{t=1}^{n} \mathrm{I}_t(\nu_m)\, P(X_t = j | \, \mathbf{S}, \lambda^*)}{\beta}.
\end{aligned}$$

So, from the above,

$$\begin{aligned}
1 = \sum_{v_m \in \delta} \hat{b}_{jm} &= -\frac{1}{\beta} \sum_{v_m \in \delta} \sum_{t=1}^{n} \mathrm{I}_t(\nu_m)\, P(X_t = j | \, \mathbf{S}, \lambda^*) \\
&= -\frac{1}{\beta} \sum_{t=1}^{n} P(X_t = j | \, \mathbf{S}, \lambda^*) \sum_{v_m \in \delta} \mathrm{I}_t(\nu_m) \\
&= -\frac{1}{\beta} \sum_{t=1}^{n} P(X_t = j | \, \mathbf{S}, \lambda^*)\, (1) \\
\Rightarrow \beta &= -\sum_{t=1}^{n} P(X_t = j | \, \mathbf{S}, \lambda^*) \\
\Rightarrow \hat{b}_{jm} &= \frac{\sum_{t=1}^{n} \mathrm{I}_t(\nu_m)\, P(X_t = j | \, \mathbf{S}, \lambda^*)}{\sum_{t=1}^{n} P(X_t = j | \, \mathbf{S}, \lambda^*)} \\
&= \frac{\sum_{t=1}^{n} \gamma_t^*(j)\, \mathrm{I}_t(\nu_m)}{\sum_{t=1}^{n} \gamma_t^*(j)} \quad \text{..... by (4.3) and (4.6)}
\end{aligned}$$

$$= \frac{\sum_{t=1}^{n} \gamma^*_{t,m}(j)}{\sum_{t=1}^{n} \gamma^*_t(j)} \quad \text{..... by (4.3) and (4.6).}$$

And so the estimate for b_{jm} obtained from the BWA (see equation (4.9)) is the same as the above MLE of b_{jm} obtained using the EM algorithm.

Using similar techniques to that which was used for $\hat{p}_i$, it can be verified that $\hat{b}_{jm}$ is indeed a local maxima.

And so the BWA estimates for p_i, p_{ij} and b_{jm} are indeed identical to the MLEs of these parameters obtained when the EM algorithm is applied to the HMM. That is the Baum-Welch re-estimation equations (equations (4.7)-(4.9)) are essentially identical to the iteration steps of the EM algorithm described above.

To summarise, the above defined $\hat{\lambda}$ is the value for λ which will maximise the function $Q(\lambda, \lambda^*)$. From the discussion in Section B.2.1, this implies that $L(\hat{\lambda}) \geq L(\lambda^*)$, or equivalently that $P(\mathbf{S} = (s_1, s_2, \ldots s_n) | \, \hat{\lambda}) \geq P(\mathbf{S} = (s_1, s_2, \ldots s_n) | \, \lambda^*)$. And so the likelihood function will continually be increased with each iteration until convergence to a critical point of the likelihood function is reached.

The EM Algorithm for a HMM with a stationary Markov Chain

For certain applications of the HMM it may be desirable to assume that the underlying Markov chain is stationary. Recall from equation (1.10) that under this assumption

$$\mathbf{a} = \mathbf{1}(\mathbf{I}_m - \mathbf{P} + \mathbf{U}_m)^{-1},$$

where it is arbitrarily assumed that there are m states in the state space and that $\mathbf{a}$ is the m-dimensional row vector containing the initial probabilities $p_i = P(X_1 = i)$ for each $i \in \{1, 2, \ldots, m\}$. From the above it can be seen that $\mathbf{a}$ is completely determined by the transition probabilities contained in $\mathbf{P}$, and therefore the question of estimating $\mathbf{a}$ falls away. However, in determining the MLEs for the transition probabilities the

M step gives rise to the following maximisation problem: maximise, with respect to $\mathbf{P}$, the first two terms of $Q(\lambda, \lambda^*)$. That is for each $k, l \in \alpha$ maximise

$$\sum_{k\in\alpha} \ln(p_k)\, P(X_1 = k|\, \mathbf{S}, \lambda^*) + \sum_{k\in\alpha} \sum_{l\in\alpha} \ln(p_{k,l}) \sum_{t=1}^{n-1} P(X_t = k, X_{t+1} = l|\, \mathbf{S}, \lambda^*) \tag{B.10}$$

with respect to $p_{k,l}$, where the first term also depends on $\mathbf{P}$.

Even in the case of only two states, [46] points out that analytical maximisation would require the solution of a pair of quadratic equations in two variables (two of the transition probabilities), a calculation which becomes rather involved. A numerical solution is therefore typically required to perform the maximisation of (B.10) if stationarity is assumed (as is noted in [15] and [46]).

Appendix C

Additional Proofs for the Double-Chain Markov Model

Parameter estimation for the DCMM was discussed in Section 6.2 of this dissertation. In particular results were given without formal proof. This appendix will now detail these proofs. The proofs discussed were not found in the literature at the time of writing and were derived specifically for the purpose of this dissertation.

C.1 Proof of Results Used in the Baum-Welch Algorithm for the DCMM

The following interpretations were made in Section 6.2 of this dissertation when deriving the BWA estimates for the DCMM:

$$\sum_{k=1}^{n} \gamma_{k,h}(i) \quad = \quad \text{expected number of times, during the first } n \text{ time points, that the DCMM is in state } i \text{ when the previous emitted signal was } \nu_h \,,$$

$$\sum_{k=1}^{n} \gamma_{k,h,m}(i) \quad = \quad \text{expected number of times, during the first } n \text{ time points, that the DCMM is in state } i \text{ when the previous emitted signal was } \nu_h \text{ and the current signal emitted is } \nu_m \,.$$

To prove these statements recall the established statistical result which was given in equation (B.2) and is replicated again below for ease of reference:

$$\mathrm{E}(g(Z)|\, Y = y) = \sum_{z} g(z)\,.\,P(Z = z|\, Y = y)\,. \tag{C.1}$$

Also recall that $\mathbf{S}_n = (S_0, \ldots, S_n)$ is the vector of random variables for the signals emitted up to time point n, and $\mathbf{s}_n = (s_0, \ldots, s_n)$ is the actual sequence of signals which have been observed, where $s_k \in \delta$ for $k = 0, 2 \ldots, n$.

Now define W to be the number of times, during the first n time points, that the DCMM was in state i when the signal emitted at the previous time point was ν_h. Further define

$$W_{i,h,k} = \begin{cases} 1, & \text{if the DCMM is in state } i \text{ at time } k \text{ and emitted signal } \nu_h \text{ at time } k-1 \\ 0, & \text{otherwise.} \end{cases}$$

It then follows that $W = \sum_{k=1}^{n} W_{i,h,k}$ is satisfied.

Now,

Expected number of times, during the first n time points, that the DCMM is in state i when the previous emitted signal was ν_h

$$\begin{aligned} &= \mathrm{E}(W|\, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\ &= \mathrm{E}(\sum_{k=1}^{n} W_{i,h,k}|\, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\ &= \sum_{k=1}^{n} \mathrm{E}(W_{i,h,k}|\, \mathbf{S}_n = \mathbf{s}_n, \lambda) \end{aligned}$$

$$= \sum_{k=1}^{n} [(1)P(W_{i,h,k} = 1 | \mathbf{S}_n = \mathbf{s}_n, \lambda) + (0)P(W_{i,h,k} = 0 | \mathbf{S}_n = \mathbf{s}_n, \lambda)] \quad ..... \quad \text{by } (C.1)$$

$$= \sum_{k=1}^{n} P(W_{i,h,k} = 1 | \mathbf{S}_n = \mathbf{s}_n, \lambda)$$

$$= \sum_{k=1}^{n} P(X_k = i, S_{k-1} = \nu_h \,|\, \mathbf{S}_n = \mathbf{s}_n, \lambda)$$

$$= \sum_{k=1}^{n} P(X_k = i \,|\, \mathbf{S}_n = \mathbf{s}_n, \lambda)\, P(S_{k-1} = \nu_h \,|\, X_k = i, \mathbf{S}_n = \mathbf{s}_n, \lambda) \quad ..... \quad \text{by } (2.11)$$

$$= \sum_{k=1}^{n} \gamma_k(i) \times \begin{cases} 1 & \text{if} \quad s_{k-1} = \nu_h \\ 0 & \text{if} \quad s_{k-1} \neq \nu_h \end{cases} \quad ..... \quad \text{by } (6.6)$$

$$= \sum_{k=1}^{n} \begin{cases} \gamma_k(i) & \text{if} \quad s_{k-1} = \nu_h \\ 0 & \text{if} \quad s_{k-1} \neq \nu_h \end{cases}$$

$$= \sum_{k=1}^{n} \gamma_{k,h}(i) \quad ..... \quad \text{by } (6.6).$$

And so the first expected value result is proven.

Next define Y to be the number of times, during the first n time points, that the DCMM was in state i when the signal emitted at the previous time point was ν_h and the current signal emitted is ν_m . Further define

$$Y_{i,h,m,k} = \begin{cases} 1, & \text{if the DCMM is in state } i \text{ at time } k \text{ and emitted signal } \nu_h \text{ at time} \\ & k-1 \text{ and emitted signal } \nu_m \text{ at time } k \\ 0, & \text{otherwise.} \end{cases}$$

It then follows that $Y = \sum_{k=1}^{n} Y_{i,h,m,k}$ is satisfied.

And so, using similar mathematics as above, it follows that

Expected number of times, during the first n time points, that the DCMM is in state i when the previous emitted signal was ν_h and the current emitted signal is ν_m

$$= \mathrm{E}(Y | \mathbf{S}_n = \mathbf{s}_n, \lambda)$$

$$
\begin{aligned}
&= \mathrm{E}(\sum_{k=1}^{n} Y_{i,h,m,k} |\, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} P(Y_{i,h,m,k} = 1 |\, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} P(X_k = i, S_{k-1} = \nu_h, S_k = \nu_m \,|\, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&= \sum_{k=1}^{n} P(X_k = i |\, \mathbf{S}_n = \mathbf{s}_n, \lambda)\, P(S_{k-1} = \nu_h, S_k = \nu_m \,|\, X_k = i, \mathbf{S}_n = \mathbf{s}_n, \lambda) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{.....} \quad \text{by (2.11)} \\
&= \sum_{k=1}^{n} \gamma_k(i) \times \begin{cases} 1 & \text{if} \quad s_{k-1} = \nu_h \text{ and } s_k = \nu_m \\ 0 & \text{otherwise} \end{cases} \qquad \text{.....} \quad \text{by (6.6)} \\
&= \sum_{k=1}^{n} \begin{cases} \gamma_k(i) & \text{if} \quad s_{k-1} = \nu_h \text{ and } s_k = \nu_m \\ 0 & \text{otherwise} \end{cases} \\
&= \sum_{k=1}^{n} \gamma_{k,h,m}(i) \quad \text{.....} \quad \text{by (6.6).}
\end{aligned}
$$

And so the second expected value result is proven.

C.2 Using the EM Algorithm to Estimate the Parameters of a DCMM

It was stated in Section 6.2 that the Baum-Welch algorithm (BWA) estimates for the DCMM can be derived through the use of the Expected Maximization (EM) algorithm (in particular the BWA estimates were considered for the discrete-time, discrete-state and discrete-signal DCMM where the state transition probability matrix and the signal transition probability matrix for each state are assumed time homogeneous). This section will prove the result. It should be noted that a large portion of the mathematical content of this section was not found in any references within the literature and was derived specifically for the purpose of adding clarity to this dissertation.

Recall from Appendix B that the EM algorithm is an iterative procedure which can

be used to find the maximum likelihood estimate (MLE) of parameters which depend on unobserved or latent variables. The EM algorithm is therefore of interest as typically in applications of the DCMM it will be required to estimate the DCMM model parameters without having observed the state sequence. Furthermore, if it can be shown that the BWA estimates for the DCMM are equivalent to those derived from the EM algorithm, then the appealing property that estimates derived from the EM algorithm are in fact MLEs will also extend to the BWA estimates. That is, the BWA estimates will have the favourable properties of being both computationally compact (see Section 6.2) while still giving rise to MLEs. It is however important to recall from Appendix B that estimates derived from the EM algorithm result in local maximization of the likelihood, but not necessary global maximization.

It will also be shown in this appendix that Lagrange multipliers ensure that the derived parameter estimates satisfy the following important properties: $\sum_{i \in S} \hat{p}_i = 1$, $\sum_{j \in S} \hat{p}_{ij} = 1$, and $\sum_{v_l \in \delta} \hat{b}_{jl}^{(i)} = 1$, where $i \in S$ and $v_j \in \delta$.

To begin, recall from Appendix B that each iteration of the EM algorithm can be performed by making use of the following:

$$\begin{aligned} \theta_{n+1} &= \arg\max_{\theta} \{Q(\theta, \theta_n)\} \\ &= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} P(\mathbf{z}|\, \mathbf{X}, \theta_n) \ln P(\mathbf{X}, \mathbf{z}|\, \theta) \right\}, \end{aligned} \tag{C.2}$$

where $\mathbf{X}$ represents the observed data, $\mathbf{z}$ represents the unobserved data and θ_n represents the n^{th} iteration estimate of the parameter set θ $(n = 1, 2, \ldots)$.

In order to ease the computations which are to follow (while as far as possible keeping the notation consistent to that which has been previously used in this dissertation),

consider the following notation:

λ denote the set of unknown parameters for the DMM which need to be estimated,

λ^* denote the current estimate of λ,

$\hat{\lambda}$ denote the updated estimate of λ,

$\mathbf{X} = (X_1\,, X_2\,, \ldots\,, X_n)$ be the random vector denoting the hidden states visited by the DCMM during the first n time points (previously notated as $\mathbf{X}_n$),

$\mathbf{x} = (i_1\,, i_2\,, \ldots\,, i_n)$ be a realisation of $\mathbf{X}$,

$\mathbf{S} = (S_0\,, S_1\,, \ldots\,, S_n)$ be the random vector denoting the signals observed up to time point n (previously notated as $\mathbf{S}_n$),

$\mathbf{s} = (s_0\,, s_1\,, \ldots\,, s_n)$ be a realisation of $\mathbf{S}$,

α denote the state space (previously notated as S),

δ denote the signal space.

Using the above notation, equation (C.2) is written for the DCMM as

$$\begin{aligned} \hat{\lambda} &= \arg\max_{\lambda} \{Q(\lambda, \lambda^*)\} \\ &= \arg\max_{\lambda} \left\{ \sum_{\mathbf{x}\in\mathbf{X}} P(\mathbf{x}|\,\mathbf{S}, \lambda^*) \ln P(\mathbf{S}, \mathbf{x}|\,\lambda) \right\}, \end{aligned} \tag{C.3}$$

where

$$\sum_{\mathbf{x}\in\mathbf{X}} P(\mathbf{x}) \equiv \sum_{i_1\in\alpha} \sum_{i_2\in\alpha} \cdots \sum_{i_n\in\alpha} P(\mathbf{X} = (i_1, i_2, \ldots, i_n))\,.$$

Through the use of equation (6.5), the following can be derived for the DCMM:

$$\ln P(\mathbf{S}, \mathbf{x}|\,\lambda) = \ln(p_{i_1}) + \sum_{t=1}^{n-1} \ln(p_{i_t, i_{t+1}}) + \sum_{t=1}^{n} \ln(b^{(i_t)}_{s_{t-1}, s_t})\,.$$

Using this result, function $Q(\lambda, \lambda^*)$ can then be written as follows:

$$\begin{aligned} Q(\lambda, \lambda^*) = \quad & \sum_{\mathbf{x} \in \mathbf{X}} \ln(p_{i_1}) P(\mathbf{x}|\, \mathbf{S}, \lambda^*) + \sum_{\mathbf{x} \in \mathbf{X}} \sum_{t=1}^{n-1} \ln(p_{i_t, i_{t+1}}) P(\mathbf{x}|\, \mathbf{S}, \lambda^*) \\ & + \sum_{\mathbf{x} \in \mathbf{X}} \sum_{t=1}^{n} \ln(b^{(i_t)}_{s_{t-1}, s_t}) P(\mathbf{x}|\, \mathbf{S}, \lambda^*)\,. \end{aligned} \tag{C.4}$$

Now, using similar techniques to those which were described for the HMM in Section B.2.2, the first two terms of equation (C.4) can be expressed as follows:

$$\begin{aligned} \sum_{\mathbf{x} \in \mathbf{X}} \ln(p_{i_1}) P(\mathbf{x}|\, \mathbf{S}, \lambda^*) &= \sum_{k \in \alpha} \ln(p_k)\, P(X_1 = k|\, \mathbf{S}, \lambda^*) \\ \sum_{\mathbf{x} \in \mathbf{X}} \sum_{t=1}^{n-1} \ln(p_{i_t, i_{t+1}}) P(\mathbf{x}|\, \mathbf{S}, \lambda^*) &= \sum_{k \in \alpha} \sum_{l \in \alpha} \ln(p_{k,l}) \sum_{t=1}^{n-1} P(X_t = k, X_{t+1} = l|\, \mathbf{S}, \lambda^*)\,. \end{aligned}$$

The final term of equation (C.4) can be simplified as follows:

$$\begin{aligned} & \sum_{\mathbf{x} \in \mathbf{X}} \sum_{t=1}^{n} \ln(b^{(i_t)}_{s_{t-1}, s_t}) P(\mathbf{x}|\, \mathbf{S}, \lambda^*) \\ = \quad & \sum_{t=1}^{n} \sum_{i_1 \in \alpha} \cdots \sum_{i_n \in \alpha} \ln(b^{(i_t)}_{s_{t-1}, s_t}) P(X_1 = i_1, \ldots, X_n = i_n|\, \mathbf{S}, \lambda^*) \end{aligned} \tag{C.5}$$

(since the indices of the summations are not dependent on each other, the order of summation may be changed).

Now, assume that the index for t is currently at the value $t^* \in \{1, 2, \ldots, n\}$. Then for $t = t^*$, equation (C.5) becomes

$$\begin{aligned} & \sum_{i_1 \in \alpha} \cdots \sum_{i_n \in \alpha} \ln(b^{(i_{t^*})}_{s_{t^*-1}, s_{t^*}}) P(X_1 = i_1, \ldots, X_n = i_n|\, \mathbf{S}, \lambda^*) \\ = \quad & \sum_{i_{t^*} \in \alpha} \ln(b^{(i_{t^*})}_{s_{t^*-1}, s_{t^*}}) \sum_{i_1 \in \alpha} \cdots \sum_{i_{t^*-1} \in \alpha} \sum_{i_{t^*+1} \in \alpha} \cdots \sum_{i_n \in \alpha} P(X_1 = i_1, \ldots, X_n = i_n|\, \mathbf{S}, \lambda^*) \\ = \quad & \sum_{i_{t^*} \in \alpha} \ln(b^{(i_{t^*})}_{s_{t^*-1}, s_{t^*}})\; P(X_{t^*} = i_{t^*}|\, \mathbf{S}, \lambda^*) \quad \text{.....} \quad \text{by (2.9)} \\ = \quad & \sum_{k \in \alpha} \ln(b^{(k)}_{s_{t^*-1}, s_{t^*}})\; P(X_{t^*} = k|\, \mathbf{S}, \lambda^*)\,. \end{aligned}$$

And so the final term of equation (C.4) can be written as

$$\begin{aligned} & \sum_{t=1}^{n} \sum_{k\in\alpha} \ln(b^{(k)}_{s_{t-1},s_t})\ P(X_t = k|\,\mathbf{S},\lambda^*) \\ = \quad & \sum_{k\in\alpha} \sum_{t=1}^{n} \ln(b^{(k)}_{s_{t-1},s_t})\ P(X_t = k|\,\mathbf{S},\lambda^*)\,. \end{aligned}$$

Using the above, the function $Q(\lambda,\lambda^*)$ has been simplified to the following:

$$\begin{aligned} Q(\lambda,\lambda^*) = \quad & \sum_{k\in\alpha} \ln(p_k)\, P(X_1 = k|\,\mathbf{S},\lambda^*) \\ & + \sum_{k\in\alpha} \sum_{l\in\alpha} \ln(p_{k,l}) \sum_{t=1}^{n-1} P(X_t = k, X_{t+1} = l|\,\mathbf{S},\lambda^*) \\ & + \sum_{k\in\alpha} \sum_{t=1}^{n} \ln(b^{(k)}_{s_{t-1},s_t})\ P(X_t = k|\,\mathbf{S},\lambda^*)\,. \end{aligned}$$

Recall from equation (C.3) that

$$\hat{\lambda} = \arg\max_{\lambda} \{Q(\lambda,\lambda^*)\}\,.$$

In particular it is desired the parameters of $\hat{\lambda}$ be found subject to

$$\begin{aligned} & \sum_{i\in\alpha} \hat{p}_i = 1 \\ & \sum_{j\in\alpha} \hat{p}_{ij} = 1, \qquad \text{for } i \in \alpha \\ & \sum_{v_l\in\delta} \hat{b}^{(i)}_{jl} = 1, \qquad \text{for } i \in \alpha \text{ and } v_j \in \delta\,. \end{aligned}$$

This can be achieved by making use of the Lagrange multiplier and setting the partial derivative to zero.

For $i \in \alpha$, the MLE for p_i can be derived through solving the following:

$$\begin{aligned} \frac{\partial}{\partial p_i}\left[Q(\lambda,\lambda^*) + \beta(\sum_{k\in\alpha} p_k - 1)\right] &= 0 \\ \frac{\partial}{\partial p_i}\left[\sum_{k\in\alpha} \ln(p_k)\, P(X_1 = k|\,\mathbf{S},\lambda^*)\right] + \frac{\partial}{\partial p_i}\left[\beta(\sum_{k\in\alpha} p_k - 1)\right] &= 0\,. \end{aligned}$$

Using similar techniques to those used for the HMM in Appendix B, the solution of the above yields

$$\hat{p}_i = P(X_1 = i | \, \mathbf{S}, \lambda^*) = \gamma_1^*(i) \quad ..... \quad \text{by (6.6) and (6.7).}$$

Evaluating the second partial derivative at the above calculated $\hat{p}_i$ yields that $\hat{p}_i = \gamma_1^*(i)$ is indeed a local maxima. And so it is proven that, for the DCMM, the BWA estimate for p_i (see equation (6.8)) is indeed consistent with the MLE derived using the EM algorithm.

Next, for $i, j \in \alpha$, the MLE for p_{ij} can be derived by solving the following:

$$\frac{\partial}{\partial p_{ij}} \left[Q(\lambda, \lambda^*) + \beta(\sum_{k \in \alpha} p_{ik} - 1) \right] = 0$$

$$\frac{\partial}{\partial p_{ij}} \left[\sum_{k \in \alpha} \sum_{l \in \alpha} \ln(p_{k,l}) \sum_{t=1}^{n-1} P(X_t = k, X_{t+1} = l | \, \mathbf{S}, \lambda^*) \right] + \frac{\partial}{\partial p_{ij}} \left[\beta(\sum_{k \in \alpha} p_{ik} - 1) \right] = 0 \, .$$

Using similar techniques to those used for the HMM in Appendix B, the solution of the above yields

$$\hat{p}_{ij} = \frac{\sum_{t=1}^{n-1} P(X_t = i, X_{t+1} = j | \, \mathbf{S}, \lambda^*)}{\sum_{t=1}^{n-1} P(X_t = i | \, \mathbf{S}, \lambda^*)} = \frac{\sum_{t=1}^{n-1} \xi_t^*(i,j)}{\sum_{t=1}^{n-1} \gamma_t^*(i)} \quad ..... \quad \text{by (6.6) and (6.7).}$$

Evaluating the second partial derivative at the above calculated $\hat{p}_{ij}$ yields that $\hat{p}_{ij} = \sum_{t=1}^{n-1} \xi_t^*(i,j) \, / \sum_{t=1}^{n-1} \gamma_t^*(i)$ is indeed a local maxima. And so it is proven that, for the DCMM, the BWA estimate for p_{ij} (see equation (6.9)) is indeed consistent with the MLE derived using the EM algorithm.

Next, for $j \in \alpha$ and $\nu_l, \nu_m \in \delta$ the MLE for $b_{lm}^{(j)}$, subject to the constraint $\sum_{v_h \in \delta} \hat{b}_{lh}^{(j)} = 1$, will be derived. To begin, for $t \in \{1, 2, \ldots, n\}$, define $\mathrm{J}_t(l)$ and $\mathrm{I}_t(l,m)$ to be the

following indicator functions

$$\begin{aligned} \mathrm{J}_t(l) &= \begin{cases} 1 & \text{if } s_{t-1} = \nu_l \\ 0 & \text{otherwise} \end{cases} \\ \mathrm{I}_t(l,m) &= \begin{cases} 1 & \text{if } s_{t-1} = \nu_l \text{ and } s_t = \nu_m \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

An important result regarding these indicators, which will be called upon later in the derivation of $\hat{b}_{lm}^{(j)}$, is

$$\sum_{\nu_m \in \delta} \mathrm{I}_t(l,m) = \mathrm{J}_t(l)\,. \tag{C.6}$$

To see this arbitrarily assume that there are M signals in signal space, that is $\delta = \{\nu_1, \nu_2, \ldots, \nu_M\}$. Then

$$\begin{aligned} \sum_{\nu_m \in \delta} \mathrm{I}_t(l,m) &= \mathrm{I}_t(l,1) + \mathrm{I}_t(l,2) + \ldots + \mathrm{I}_t(l,M) \\ &= \begin{cases} 1 & \text{if } s_{t-1} = \nu_l \text{ and } s_t = \nu_1 \\ 0 & \text{otherwise} \end{cases} + \ldots + \begin{cases} 1 & \text{if } s_{t-1} = \nu_l \text{ and } s_t = \nu_M \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } s_{t-1} = \nu_l \\ 0 & \text{otherwise} \end{cases} \\ &= \mathrm{J}_t(l)\,. \end{aligned}$$

Now the MLE for $b_{lm}^{(j)}$ can be derived by solving the following

$$\begin{aligned} \frac{\partial}{\partial b_{lm}^{(j)}} \left[Q(\lambda, \lambda^*) + \beta(\sum_{v_h \in \delta} b_{lh}^{(j)} - 1) \right] &= 0 \\ \frac{\partial}{\partial b_{lm}^{(j)}} \left[\sum_{k \in \alpha} \sum_{t=1}^{n} \ln(b_{s_{t-1},s_t}^{(k)})\, P(X_t = k | \,\mathbf{S}, \lambda^*) \right] + \frac{\partial}{\partial b_{lm}^{(j)}} \left[\beta(\sum_{v_h \in \delta} b_{lh}^{(j)} - 1) \right] &= 0\,. \end{aligned} \tag{C.7}$$

Notice that

$$\begin{aligned}
\frac{\partial}{\partial b_{lm}^{(j)}}\left[\ln(b_{s_{t-1},s_t}^{(k)})\right] &= 0 \qquad\qquad \forall\, k \neq j \\
\frac{\partial}{\partial b_{lm}^{(j)}}\left[\ln(b_{s_{t-1},s_t}^{(j)})\right] &= \begin{cases} \frac{1}{b_{lm}^{(j)}} & \text{if } s_{t-1}=\nu_l \text{ and } s_t=\nu_m \\ 0 & \text{otherwise} \end{cases} \\
\frac{\partial}{\partial b_{lm}^{(j)}}\left[b_{lh}^{(j)}\right] &= \begin{cases} 1 & \text{if } \nu_h=\nu_m \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Using the above, equation (C.7) can be simplified as follows

$$\begin{aligned}
\sum_{t=1}^{n} \frac{1}{b_{lm}^{(j)}} \mathrm{I}_t(l,m)\, P(X_t=j|\,\mathbf{S},\lambda^*) + \beta &= 0 \\
\Rightarrow\ \hat{b}_{lm}^{(j)} &= -\frac{\sum_{t=1}^{n} \mathrm{I}_t(l,m)\, P(X_t=j|\,\mathbf{S},\lambda^*)}{\beta}\,.
\end{aligned}$$

And so

$$\begin{aligned}
1=\sum_{\nu_m\in\delta} \hat{b}_{lm}^{(j)} &= -\frac{1}{\beta}\sum_{\nu_m\in\delta}\sum_{t=1}^{n} \mathrm{I}_t(l,m)\, P(X_t=j|\,\mathbf{S},\lambda^*) \\
&= -\frac{1}{\beta}\sum_{t=1}^{n} P(X_t=j|\,\mathbf{S},\lambda^*)\sum_{\nu_m\in\delta} \mathrm{I}_t(l,m) \\
&= -\frac{1}{\beta}\sum_{t=1}^{n} P(X_t=j|\,\mathbf{S},\lambda^*)\,\mathrm{J}_t(l) \quad \text{.....} \quad \text{by } (C.6) \\
\Rightarrow\ \beta &= -\sum_{t=1}^{n} \mathrm{J}_t(l)\, P(X_t=j|\,\mathbf{S},\lambda^*)
\end{aligned}$$

$$
\begin{aligned}
\Rightarrow\ \hat{b}_{lm}^{(j)} &= \frac{\sum_{t=1}^{n} \mathrm{I}_t(l,m)\, P(X_t = j \,|\, \mathbf{S}, \lambda^*)}{\sum_{t=1}^{n} \mathrm{J}_t(l)\, P(X_t = j \,|\, \mathbf{S}, \lambda^*)} \\
&= \frac{\sum_{t=1}^{n} \mathrm{I}_t(l,m)\, \gamma_t^*(j)}{\sum_{t=1}^{n} \mathrm{J}_t(l)\, \gamma_t^*(j)} \quad \text{..... by (6.6) and (6.7)} \\
&= \frac{\sum_{t=1}^{n} \gamma_{t,l,m}^*(j)}{\sum_{t=1}^{n} \gamma_{t,l}^*(j)} \quad \text{..... by (6.6) and (6.7).}
\end{aligned}
$$

Evaluating the second partial derivative at the above calculated $\hat{b}_{lm}^{(j)}$ yields that $\hat{b}_{lm}^{(j)} = \sum_{t=1}^{n} \gamma_{t,l,m}^*(j) \,/\, \sum_{t=1}^{n} \gamma_{t,l}^*(j)$ is indeed a local maxima. And so it is proven that, for the DCMM, the BWA estimate for $b_{lm}^{(j)}$ (see equation (6.10)) is indeed consistent with the MLE derived using the EM algorithm.

For completeness, the second partial derivative is shown below:

$$
\begin{aligned}
& \frac{\partial^2}{\partial {b_{lm}^{(j)}}^2} \left[Q(\lambda, \lambda^*) + \beta \Big(\sum_{v_h \in \delta} b_{lh}^{(j)} - 1 \Big) \right] \\
= \ & \frac{\partial}{\partial b_{lm}^{(j)}} \left[\sum_{t=1}^{n} \frac{1}{b_{lm}^{(j)}}\, \mathrm{I}_t(l,m)\, P(X_t = j \,|\, \mathbf{S}, \lambda^*) + \beta \right] \\
= \ & \frac{\partial}{\partial b_{lm}^{(j)}} \left[\frac{1}{b_{lm}^{(j)}} \sum_{t=1}^{n} \gamma_{t,l,m}^*(j) + \beta \right] \\
= \ & -\frac{1}{[b_{lm}^{(j)}]^2} \sum_{t=1}^{n} \gamma_{t,l,m}^*(j)\,.
\end{aligned}
$$

And so,

$$\begin{aligned}
& \frac{\partial^2}{\partial {b_{lm}^{(j)}}^2} \left[Q(\lambda, \lambda^*) + \beta(\sum_{v_h \in \delta} b_{lh}^{(j)} - 1) \right] \Bigg|_{b_{lm}^{(j)} = \hat{b}_{lm}^{(j)}} \\
= & - \left(\frac{\sum_{t=1}^{n} \gamma_{t,l}^*(j)}{\sum_{t=1}^{n} \gamma_{t,l,m}^*(j)} \right)^2 \sum_{t=1}^{n} \gamma_{t,l,m}^*(j) \\
= & - \frac{\left(\sum_{t=1}^{n} \gamma_{t,l}^*(j) \right)^2}{\sum_{t=1}^{n} \gamma_{t,l,m}^*(j)} .
\end{aligned}$$

References

[1] Anderson, T.W. and Goodman, L.A. *Statistical Inference about Markov Chains.* The Annals of Mathematical Sciences, vol. 28, no.1, pp. 89-110. 1957.

[2] Andriyas, S. and McKee, M. *Exploring Irrigation Behavior at Delta, Utah using Hidden Markov Models.* Agricultural Water Management, vol. 143, pp. 48-58. 2014.

[3] Banachewicz, K. and Lucas, A. *Quantile Forecasting for Credit Risk Management using Possibly Misspecified Hidden Markov Models.* Journal of Forecasting, vol. 27, no. 7, pp. 566-586. 2008

[4] Banachewicz, K., Lucas, A. and van der Vaart, A. *Modelling Portfolio Defaults using Hidden Markov Models with Covariates.* The Econometrics Journal, vol. 11, no.1, pp. 155-171. 2008.

[5] Baum, L.E. *An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes.* Inequalities, vol. 3, pp. 1-8. 1972

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

[7] Baum, L.E. and Petrie, T. *Statistical Inference for Probabilistic Functions of Finite State Markov Chains.* The Annals of Mathematical Statistics, vol. 37, pp. 1554-1563. 1966

[8] Baum, L.E. and Sell, G.R. *Growth Transformations for Functions on Manifolds.* Pacific Journal of Mathematics, vol. 27, no. 2, pp. 211-227. 1968.

[9] Baum, L.E., Petrie, T., Soules, G. and Weiss, N. *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains.* The Annals of Mathematical Statistics, vol. 41, pp. 164-171. 1970.

[10] Berchtold, A. *The Double Chain Markov Model.* Communications in Statistics - Theory and Methods, vol. 28, no. 11, pp. 2569-2589. 1999.

[11] Berchtold, A. *Higher-Order Extensions of the Double Chain Markov Model.* Stochastic Models, vol. 18, no. 2, pp.193-227. 2002.

[12] Bilmes, J.A. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.* International Computer Science Institute, Technical Report ICSI-TR-97-021. 1998.

[13] Borman, S. *The Expectation Maximization Algorithm: A Short Tutorial.* Technical Report. 2004.
URL : http://www.cs.utah.edu/ piyush/teaching/EM_algorithm.pdf

[14] Bracken, C., Rajagopalan, B. and Zagona, E. *A Hidden Markov Model Combined with Climate Indices for Multidecadal Streamflow Simulation.* Water Resources Research, vol. 50, no. 10, pp. 7836-7846. 2014

[15] Bulla, J. and Berzel, A. *Computational Issues in Parameter Estimation for Stationary Hidden Markov Models.* Computational Statistics, vol. 23, no. 1, pp.1-18. 2008.

[16] Can, C.E., Ergun, G. and Gokceoglu, C. *Prediction of Earthquake Hazard by Hidden Markov Model (around Bilecik, NW Turkey).* Central European Journal of Geosciences, vol. 6, no. 3, pp. 403-414. 2014.

[17] Casella, G. and Berger, R.L. *Statistical Inference,* 2^{nd} *edition.* Duxbury Thomson Learning, University of Florida and North Carolina State University. 2002.

[18] Davis, R.I.A., Lovell, B.C. and Caelli, T. *Improved Estimation of Hidden Markov Model Parameters from Multiple Observation Sequences.* Proceedings of 16th International Conference on Pattern Recognition, vol. 2, pp. 168-171. 2002.

[19] Davison, A.C. *Statistical Models.* Cambridge University Press, Cambridge. 2003.

[20] Dempster, A.P., Laird, N.M. and Rubin, D.B. *Maximum Likelihood from Incomplete Data via the EM Algorithm.* Journal of the Royal Statistical Society, series B, vol. 39, no. 1, pp. 1-38. 1977.

[21] Efron, B. and Tibshirani, R.J. *An Introduction to the Bootstrap.* Chapman and Hall / CRC Press, Stanford University and University of Toronto. 1993.

[22] Eisenkopf, A. *The Real Nature of Credit Rating Transitions.* Goethe University Frankfurt. 2008.
URL : http://ssrn.com/abstract=968311

[23] Fitzpatrick, M. and Marchev, D. *Efficient Bayesian Estimation of the Multivariate Double Chain Markov Model.* Statistics and Computing, vol. 23, no. 4, pp. 467-480. 2013.

[24] Forney, G.D. *The Viterbi Algorithm.* Proceedings of the IEEE, vol. 61, no. 3, pp. 268-278. 1973.

[25] Forney, G.D. *The Viterbi Algorithm: A Personal History.* 2005.
URL : http://arxiv.org/abs/cs.IT/0504020

[26] Fort, A., Mugnaini, M. and Vignoli, V. *Hidden Markov Models Approach used for Life Parameters Estimations.* Reliability Engineering and System Safety, vol. 136, pp. 85-91. 2015.

[27] Giampieri, G., Davis, M. and Crowder, M. *Analysis of Default Data using Hidden Markov Models.* Quantitative Finance, vol. 5, no.1, pp.27-34. 2005.

[28] Gomez, A.E., Alencar, F.A.R., Prado, P.V., Osorio, F.S. and Wolf, D.F. *Traffic Lights Detection and State Estimation Using Hidden Markov Models.* Proceedings of IEEE Intelligent Vehicles Symposium, pp. 750-755. 2014.

[29] Henderson, J., Salzberg S. and Fasman, K.H. *Finding Genes in DNA with a Hidden Markov Model.* Journal of Computational Biology, vol. 4, no. 2, pp. 127-141. 2009.

[30] Korolkiewicz, M.W. and Elliott, R.J. *A Hidden Markov Model of Credit Quality.* Journal of Economic Dynamics and Control, vol. 32, no. 12, pp. 3807-3819. 2008.

[31] Levinson, S.E., Rabiner, L.R. and Sondhi, M.M. *An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition.* The Bell System Technical Journal, vol. 62, no. 4, pp. 1035-1074. 1983.

[32] Li, X., Parizeau, M. and Plamondon, R. *Training Hidden Markov Models with Multiple Observations - A Combinatorial Method.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 4, pp. 371-377. 2000.

[33] Lou, H.L. *Implementing the Viterbi Algorithm.* IEEE Signal Processing Magazine, vol. 12, no. 5, pp. 42-52. 1995.

[34] MacDonald, I.L. *Numerical Maximisation of Likelihood: A Neglected Alternative to EM?* International Statistical Review, vol. 82, no. 2, pp. 296-308. 2014.

[35] MacDonald, I.L. and Zucchini, W. *Hidden Markov and Other Models for Discrete-valued Time Series.* Chapman & Hall, University of Cape Town and University of Göttingen. 1997.

[36] McLachlan, G.J. and Krishnan, T. *The EM Algorithm and Extensions,* 2^{nd} *edition.* John Wiley and Sons, University of Queensland. 2008.

[37] Rabiner, L.R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286. 1989.

[38] Rabiner, L.R. and Juang, B.H. *An Introduction to Hidden Markov Models.* IEEE ASSP Magazine, vol. 3, no. 1, pp. 4-16. 1986.

[39] Raftery, A.E. *A Model for High-order Markov Chains.* Journal of the Royal Statistical Society, series B, vol. 47, no. 3, pp. 528-539. 1985.

[40] Resch, B. *Hidden Markov Models, A Tutorial for the Course Computational Intelligence.* Signal Processing and Speech Communication Laboratory. 2004.

[41] Ross, S.M. *Introduction to Probability Models,* 9^{th} *edition.* Elsevier, University of California. 2007.

[42] Satish, L. *Use of Hidden Markov Models for Partial Discharge Pattern Classification.* IEEE Transactions of Electrical Insulation, vol. 28, no. 2, pp.172-182. 1993.

[43] Starner, T. and Pentland A. *Real-time American Sign Language Recognition from Video using Hidden Markov Models.* Proceedings of International Symposium on Computer Vision, pp. 265-270. 1995.

[44] Viterbi, A.J. *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.* IEEE Transactions on Information Theory, vol. 13, no. 2, pp. 260-269. 1967.

[45] Yoo, J., Kwon, H.H., So, B.J., Rajagopalan, B. and Kim, T.W. *Identifying the Role of Typhoons as Drought Busters in South Korea based on Hidden Markov Chain Models.* Geophysical Research Letters, vol. 42, no. 8, pp. 2797-2804. 2015.

[46] Zucchini, W. and MacDonald, I.L. *Hidden Markov Models for Time Series - An Introduction Using R.* Chapman & Hall / CRC Press, Georg-August-Universität and University of Cape Town. 2009.